

Attorney Docket Number 53470.003013

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of: )  
 )  
Jeffrey A. BEDELL, et al. )  
 )  
Serial No.: 09/883,475 ) Art Unit: 2134  
 )  
Filed: June 20, 2001 ) Examiner: N. Wright

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**DECLARATION OF PRIOR INVENTION UNDER 37 C.F.R. § 1.131**

Sir:

We, Jeffrey A. Bedell, Benjamin Z. Li, Luis Orozco and Ramparasad Polana, hereby declare that we are co-inventors of the invention that is claimed in the above-identified patent application. Prior to January 4, 2001, we conceived of and reduced to practice the invention that is claimed in the above-identified patent application.

Internal documentation related to the development of this feature demonstrates conception as early as January 30, 1998 with diligence through its actual reduction to practice on or after June 28, 2000. Exhibit A is a May 25, 1998 document entitled "DSS Server Job Priority" that describes one embodiment of assigning priorities and servicing them as claimed. This feature was part of a major new product development at MicroStrategy given the internal code name Castor. Castor substantially rewrote the platform of the business intelligence software system of MicroStrategy. As a result, this feature, and many others, went through numerous rounds of tests, a confidential beta release and then final release on or before June of 2000 in a product called MicroStrategy 7.0. Attached as Exhibit B are various documents, including Program Review Documents and Project Schedules for the system described above. The Program Review Documents are dated from January 30, 1998 to December 23, 1999 and are

Attorney Docket Number 53470.003013

entitled "Kernel Team Milestones," "Castor Kernel Status," "Castor Server Status," and "Castor Program Status." The Project Schedules are dated from February 9, 1999 to May 21, 1999 (MS Projects) and from February 12, 2000 to May 14, 2000 ("Deliverables by Week"). The documents in Exhibit B provide evidence of diligence since they are detailed summaries of the system during product development from conception to an actual reduction to practice with the release of MicroStrategy 7.0 on or after June 28, 2000.

We further hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

DECLARANT: 

Jeffrey A. Bedell

Date: 8-9-2006DECLARANT: 

Benjamin Z. Li

Date: 8-9-2006DECLARANT: 

Luis Orozco

Date: 8/10/06DECLARANT: 

Ramparasad Polana

Date: 8/10/06

**EXHIBIT A**



# DSS Server Job Priority

<b>ROLE OF JOB PRIORITY IN JOB EXECUTION .....</b>	<b>3</b>
1.1 ENTERING JOBS INTO A QUEUE.....	3
1.2 CALCULATION OF JOB PRIORITY .....	4
<b>IMPLEMENTATION OF JOB PRIORITY .....</b>	<b>5</b>
1.3 MSIJOBPRIORITYSCHEMEOBJECT .....	5
1.4 CREATION OF MSIJOBPRIORITYSCHEMEOBJECTS .....	6
1.5 PROPOSED IMPLEMENTATION ACCORDING TO SPECIFICATIONS .....	6
1.6 CHANGING PRIORITY AFTER JOB IS ENTERED INTO QUEUE .....	8
<b>JOB SERVICING SCHEMES.....</b>	<b>9</b>
1.7 UNITS OF INDEPENDENT RESOURCE ALLOCATION AND CONTROL.....	9
1.8 JOB PROCESSING ACCORDING TO A SERVICING SCHEME .....	10
1.8.1 <i>FixedThreadCooperative</i> .....	11
1.8.2 <i>WeightedShare</i> .....	11
1.8.3 <i>HighestPriorityFirst</i> .....	12



**ABSTRACT**

*This document describes the job priority computation and usage in DSS Server which affects the execution order of user requests.*

**HISTORY**

Date	Author	Description
4/30/98	Ramprasad Polana	Initial Version
5/13/98	Ramprasad Polana	Added servicing schemes
5/25/98	Ramprasad Polana	Added changes in job priority schemes as decided through internal team review and CTA reviews

**REFERENCES:**

A must read: Castor server specification document: section on job prioritization and servicing.

## ROLE OF JOB PRIORITY IN JOB EXECUTION

DSS Server creates job objects for every user requests that can not be immediately serviced. Processing units within DSS Server execute jobs in a pipeline architecture (refer to the server internal architecture document for details on the pipeline architecture). Each processing unit contains a hierarchical queue (we will call this a station) and a pool of threads that service the queues. The jobs are placed within a queue based on its priority while the threads available to service a job will pick the first job within a queue that is selected using the servicing schemes.

Following two sections describe the process of entering the job into a particular queue of a processing unit.

### 1.1 ENTERING JOBS INTO A QUEUE

Every processing unit in DSS Server contains a hierarchical queue, which is organized as a tree. Typical processing units contain only a two-level hierarchy (except for the processing units containing the DSSQueryExecutionTask which require three-level hierarchy where the first level split by the warehouse dbc type). The leaf nodes represent basic FIFO queues, while all other nodes are a collections of queues, or queue sets.

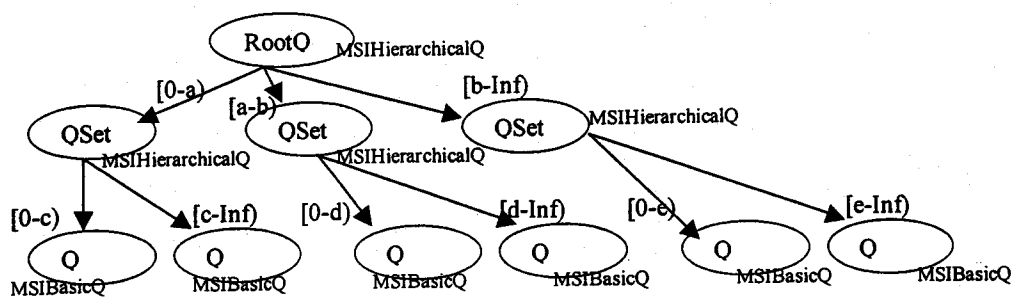


Figure 1. Hierarchical queue structure in a typical DSS Server PU. The variables a, b, c, d and e are positive integers and Inf stands for infinity.

```

MSIQRStatus MSIHierarchicalQ::Enqueue(JOB_TYPE *Job, unsigned long iMilliseconds =
gcInfiniteTimeout){
    int aPriority = mPrioritySchemeObject->CalculatePriority(Job);
    MSIQRStatus aRC;
    for(int aSubQ = 0; aSubQ < mSubQueues; aSubQ++){
        if(aPriority < mUpperBounds[aSubQ]){
            aRC = mSubQ[aSubQ] -> Enqueue(Job, iMilliseconds);
            break;
        }
    }
    if(aSubQ >= mSubQueues){
        aRC = mSubQ[mSubQueues-1] -> Enqueue(Job, iMilliseconds);
    }
    return aRC;
}
  
```

When a job needs a particular task to be done, the JobExecutor inside DSS Server finds a processing unit which can perform that task, and hands over the job to the processing unit by way of the MSIPU::Enter(MSIJob \*iJob) method. The implementation of this method in turn, calls the MSIHierarchicalQ::Enter(MSIJob \*iJob) method to place the job in a FIFO queue at a leaf within its station. This method involves computing the job priority to decide which queue or queue set should the job be placed in. At the leaf nodes, the MSIBasicQ::Enter(MSIJob \*iJob) method actually enters the job at the tail end of the FIFO queue. As shown below, at every intermediate node, each of its subqueues is associated with a range of priorities. The priority computed at intermediate nodes indicates subqueue to enter the job into.

## 1.2 CALCULATION OF JOB PRIORITY

The calculation of job priority at a QSet (MSIHierarchicalQ) node can be configured differently for different nodes to achieve different objectives in job prioritization. An example would be to split jobs at RootQ based on project and at the next level split jobs based on two user groups. With this configuration, an administrator can allocate resources such as server threads and database connections for each project and user group independently, and specify their servicing schemes as desired. This configuration can be achieved by specifying a priority function at RootQ, which ignores every priority variable other than project and returns the project id as the job priority, which will be used to index into the Qset and enter the job into the corresponding node below RootQ.

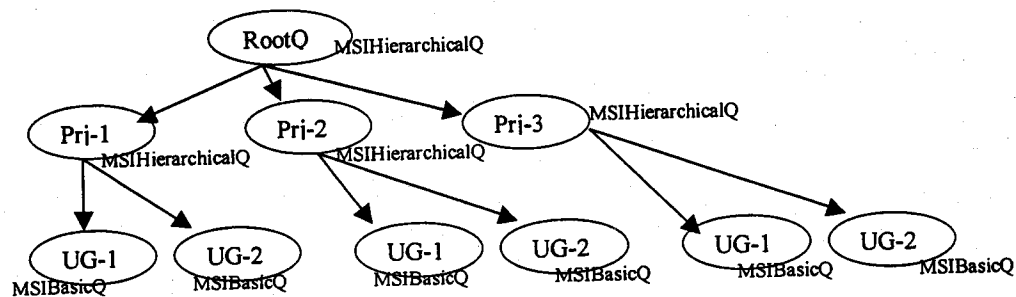


Figure 2. Configuring queue structure so that at the first level, the job is prioritized based on the project and at the second level it is prioritized based on the user group. Note that at this stage, the queue structure represents only a means of categorization. Priority ranges associated with the queues together with servicing policies determine the real priorities between queues and the order in which jobs are processed in a PU.

## IMPLEMENTATION OF JOB PRIORITY

### 1.3 MSIJOBPRIORITYSCHEMEOBJECT

To implement different priority calculations at different nodes of the hierarchical queue, every MSIHierarchical node object contains a MSIJobPrioritySchemeObject, which provides the CalculatePriority method. MSIJobPriorityScheme is an abstract base class which only provides the CalculatePriority interface method. In DSS Server, there are several classes derived from MSIJobPrioritySchemeObject, which implement the calculate priority in various ways. There are several predefined classes derived from MSIJobPrioritySchemeObject defined in the MSIPU.h as described below. For example MSIJobPrioritySchemeObjectRandom is equivalent to entering the job into one of the queue set or queue selected randomly.

```

Class ConfigManager; // forward declaration

template <class JOB_TYPE> class MSIJobPrioritySchemeObject {
protected:
    int mType;
public:
    typedef int PCFunc(JOB_TYPE *iJob);
    MSIJobPrioritySchemeObject(int iType = -1) : mType(iType) {};
    virtual int GetType() { return mType; }
    virtual ~MSIJobPrioritySchemeObject() {};
    virtual int CalculatePriority(JOB_TYPE *iJob) = 0;
    virtual bool Init(ConfigManager *iConfig, StringList & iPath) = 0;
    virtual MSIJobPrioritySchemeObject<JOB_TYPE> *Clone() = 0;
};

enum MSIJobPriorityScheme {
    gcJobPrioritySchemeDefault = 0,
    gcJobPrioritySchemeUserSupplied = 0,
    gcJobPrioritySchemeRandom,
    gcJobPrioritySchemeObjectMapBased
};

class MSIJobPrioritySchemeObjectRandom: public MSIJobPrioritySchemeObject<MSIJob> {
public:
    MSIJobPrioritySchemeObjectRandom():
        MSIJobPrioritySchemeObject<MSIJob>(gcJobPrioritySchemeRandom) {};
    int CalculatePriority(MSIJob *iJob){
        return rand();
    }
    MSIJobPrioritySchemeObject<MSIJob> *Clone() {
        return new MSIJobPrioritySchemeObjectRandom();
    };
    bool Init(ConfigManager *iConfig, StringList & iPath){
        return true;
    }
};

class MSIJobPrioritySchemeObjectUserSupplied: public MSIJobPrioritySchemeObject<MSIJob> {
public:
    MSIJobPrioritySchemeObjectUserSupplied():
        MSIJobPrioritySchemeObject<MSIJob>(gcJobPrioritySchemeUserSupplied) {};
    int CalculatePriority(MSIJob *iJob){
        return iJob->GetPriority();
    }
    MSIJobPrioritySchemeObject<MSIJob> *Clone() {
        return new MSIJobPrioritySchemeObjectUserSupplied();
    };
    bool Init(ConfigManager *iConfig, StringList & iPath){
        return true;
    }
};

```

## 1.4 CREATION OF MSIJOBPRIORITYSCHEMEOBJECTS

Note that the MSIJobPrioritySchemeObject interface also includes two more methods, namely Init and Clone methods. The Init() method is used to initialize the object with a configuration settings specific to that node. For example, this would be used to which path the object is attached to, so that any other configuration information can be obtained from the ConfigManager. The Clone() method is used by MSIPU in creation of these objects at the MSIHierarchicalQ nodes. MSIPU is given an array of predefined prototype objects, one of each type, from which MSIPU clones all other objects as the Q structure is being created.

At the DSS Server startup time, based on the configuration specified, config manager creates the processing units with the required queue structure along with the MSIJobPrioritySchemeObjects at all MSIHierarchicalQ nodes. When there is no scheme specified, either the parent node's scheme is used or a default is selected as shown below.

```
// extracted from MSIPU.cpp
MSIPU::BuildHierarchicalQ(...) {
    ...
    // check if PriorityScheme parameter is specified at this node,
    // and if so store its type in IPriorityScheme
    if(IPriorityFunctionKeyFound){
        // no PriorityFunction key found
        IPrioritySchemeObject = mPrioritySchemeObject[IPriorityScheme]->Clone();
        IPrioritySchemeObject->Init(ISubParameterTree);
    }
    else{
        // no PriorityFunction key here but this is a QSet
        if(IParentPrioritySchemeObject) {
            // parent got hold of one, clone it
            IPrioritySchemeObject = IParentPrioritySchemeObject->Clone();
            IPrioritySchemeObject->Init(ISubParameterTree);
        }
        else {
            // parent had none: probably this is the root and has no PriorityScheme specified
            IPrioritySchemeObject = mPrioritySchemeObject[gcJobPrioritySchemeDefault]->Clone();
            IPrioritySchemeObject->Init(ISubParameterTree);
        }
    }
    ...
}
```

## 1.5 PROPOSED IMPLEMENTATION ACCORDING TO SPECIFICATIONS

The above design of a queue structure with MSIJobPrioritySchemeObject at each node is general enough to implement vastly different ways of determining priorities of a job by way of overriding MSIJobPrioritySchemeObject::CalculatePriority() method in derived classes of MSIJobPrioritySchemeObject. It even allows for different ways of computing priority at different nodes, even though it is not desirable in general. DSS Server specification document fixes many of the dimensions for the sake of simplicity and manageability.

Here is a summary of the specification regarding job priorities. Job priorities are calculated based on a number of factors relevant to the job. Some of the factors are allowed to interact nonlinearly in determining the priority while others are assumed to interact linearly. DSS Server allows for the administrator to select which factors interact nonlinearly and which ones interact linearly. This is selected through a configuration wizard such as the one included in DSS Server Administrator product. Examples of factors that could interact nonlinearly are: project, user group, report type, initial report priority, time period and report cost as determined by a linear combination of other factors. The factors that are assumed to interact linearly are those that can be measured quantitatively for any job. Examples of linear factors are: historical report cost, estimated report cost, number of database queries, size of result set etc. Each linear factor is associated with a weight, and a combined report cost is computed as a weighted linear

combination of all the linear factors. This combined report cost is one of the factors allowed to interact nonlinearly in determining the job priority. The nonlinear factors determine the the job priority through a priority map (or table) whose input dimensions are all the factors allowed to interact nonlinearly and the entry in the map giving the job priority. The map representation to determine job priority is chosen because it is not only powerful enough to represent any kind of relationship between the input dimensions and the resulting priority, but also is convenient to define, store and communicate and efficient to calculate the priorities.

Thus to compute the job priority, first the values of linear factors are computed for the job, their weighted linear combination is computed as a combined report cost. Then, the priority map is looked up using the combined report cost together with all other nonlinear factors as input dimensions into the map. The map entry represents the priority for that job.

Note that the priority map is allowed to be different at different nodes of a process unit and different across process units, allowing for highly flexible priority configurations. In the current design, the nodes contain a pointer to `MSIJobPrioritySchemeObjectMapBased` object, which store a pointer to the priority map for that node. This allows for sharing the priority maps between nodes.

Also, the weights in the weighted linear combination formula are allowed to be different at different PU's. This is done by storing the multiple lists of linear factor name along with its weight in a `DSSProperty` interface in the report instance itself. Each list corresponds to a weighted linear combination formula. These lists are created in the report instance by looking up the report definition, server definition and project definition the former parameters overriding the latter in case of multiple definitions containing these parameters.

To define the priority calculation function as described above, one would specify the `PriorityScheme` parameter to the node as `gcJobPrioritySchemeObjectMapBased`. The `MSIJobPrioritySchemeObjectMapBased` class is as defined below. Individual nodes would get the map information from `ConfigManager`. `ConfigManager` would have access to all the configuration information, including the priority maps defined by the administrator at different nodes. Note that it allows the map to be shared across processing units and nodes as configured, without having to duplicate them.

// MSIConfig.h

```
class MSIJobPrioritySchemeObjectMapBased: public MSIJobPrioritySchemeObject<MSIJob> {
protected:
    ConfigManager *mConfig;    // Config has all configuration info
    StringList mPath;          // path to the SubQ including the name of the PU.
    Int mNonlinearFactors;
    Int *mNonlinearFactor;
    Int *mNonlinearFactorMaxValue;
public:
    MSIJobPrioritySchemeObjectMapBased():
        MSIJobPrioritySchemeObject<MSIJob>(gcJobPrioritySchemeObjectMapBased) {}
    int CalculatePriority(MSIJob *iJob){
        void *aPriorityMapPtr = mConfig->GetPriorityMap(mPath);
        int aValue;
        int aNonlinearFactorMaxValue = 1;
        for(int aNonlinearFactor = mNonlinearFactors-1; aNonlinearFactor >=0; aNonlinearFactor--){
            // the combined report cost is one of the nonlinear factors
            // all nonlinear factors are computed within MSIJob module
            aValue = iJob->GetFactorValue(mNonlinearFactor[aNonlinearFactor]);
            aPriorityMapPtr = (void *) ((int *)aPriorityMapPtr + aNonlinearFactorMaxValue*aValue);
            aNonlinearFactorMaxValue *= mNonlinearFactorMaxValue[aNonlinearFactor];
        }
        return *((int *)aPriorityMapPtr);
    }
    MSIJobPrioritySchemeObject<MSIJob> *Clone() {
        return new MSIJobPrioritySchemeObjectMapBased ();
    }
};
```

```
};  
bool Init(ConfigManager *iConfig, StringList & iPath){  
    mConfig = iConfig;  
    StringListCopy(mPath, iPath);  
    return true;  
}  
};
```

## 1.6 CHANGING PRIORITY AFTER JOB IS ENTERED INTO QUEUE

Once the job is entered into a queue, its priority is no longer relevant, as the servicing schemes dictate in which order jobs are serviced within a queue (servicing schemes are described in the next section). Hence, changing its priority would not affect the order in which it is serviced. If an administrator wishes to process the job earlier or later than it would have otherwise (which is usually thought of as changing the priority of a job), DSS Server allows for several commands which allow flexible job movement within or across queues.

The DSS Server commands that allow moving a job within a basic queue are:

- MoveAheadByOne: moves job ahead by one (no effect if there is no other job in front of this one)
- MoveBehindByOne: moves job behind by one (no effect if there is no job behind this one in the queue)
- MoveToFront: moves the job to front of the queue
- MoveToBack: moves the job to the back of the queue

For the above commands, the queue need not be supplied by the client, as DSS Server will find the queue in which the job is waiting and carry out the command within that queue.

The DSS Server commands that allow moving a job across queue are:

- MoveToQueue: given a job and the new queue name (which could be a hierarchical queue), DSS Server will find the queue in which the job currently is waiting, remove from there and enter it into the new queue specified.

## JOB SERVICING SCHEMES

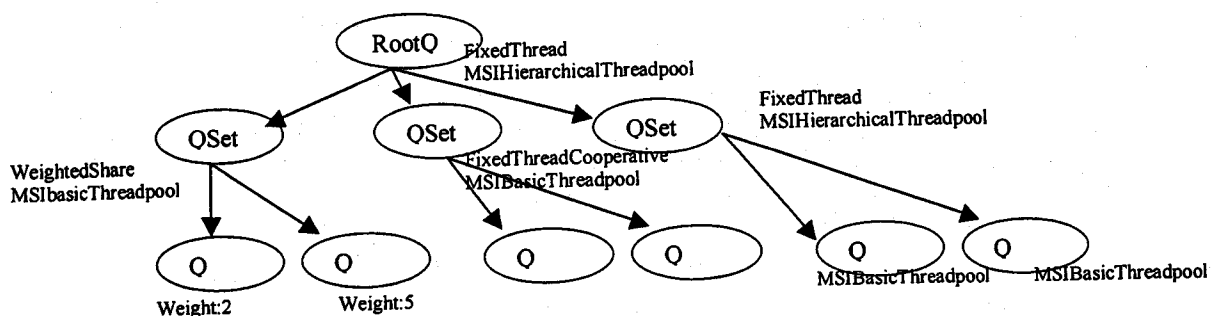
Job servicing schemes are mechanisms to control how jobs are serviced. So, the servicing scheme is applied when these resources are allocated to the jobs for their processing. That is, whenever all resources (such as server thread and database connections) become available to process a job, which job would they be allocated to next. Notice that the threads are the primary resources in a process unit, which does not do tasks involving heavy database usage. One exception, is the case of a database process unit that executes warehouse queries, where database connections (which are paired with the thread that created them) are the primary resource.

Given the fact that jobs are always serviced first-in-first-out within a queue, the job-servicing scheme in effect applies to selection of queues. That is, any time a resource is available to process a job, it is sufficient to select a queue to service next. Thus, servicing schemes are attached to a Qset, where there is a choice of queues to select from (whenever we decide to process jobs from that Qset). It follows that every Qset node in the process unit hierarchy, contains a servicing scheme specified to it. Accordingly, every MSIHierarchicalQ has a mSubQServicingScheme data member, of integer type that specifies the servicing scheme from an enumeration of available servicing schemes. The available choices of servicing schemes defined below.

```
enum MSISubQServicingScheme {
    gcSubQServicingSchemeUndefined = -1,
    gcSubQServicingSchemeDefault = 0,
    gcSubQServicingSchemeFixedThreadCooperative = 0,
    gcSubQServicingSchemeFixedThread,
    gcSubQServicingSchemeHighestPriorityFirst,
    gcSubQServicingSchemeWeightedShare
};
```

### 1.7 UNITS OF INDEPENDENT RESOURCE ALLOCATION AND CONTROL

In a process unit, the unit at which resources can be independently allocated and controlled are those Queues and Qsets that have fixed servicing schemes. A Qset that specifies a servicing scheme other than fixed implies that the subqueues within it are serviced collectively. This also implies that resources should be allocated collectively. Thus, the current design allocates independent thread pools to all Qsets that have servicing schemes other than fixed. At the Qset nodes that specify a fixed servicing scheme, the subqueues within it are serviced independently on their own, and thus have their own collection of thread pools. In this case, we associate the pool of subqueue thread pools to the Qset, to conveniently add or remove threads from the Qset. As an example, we show the servicing schemes at each node and the resulting thread pool hierarchy. MSIBasicThread pools is a collection of MSIThreads.





## 1.8 JOB PROCESSING ACCORDING TO A SERVICING SCHEME

In our process unit design, the job processing in a process unit takes place via the MSIQTask class. MSIQTask::Run() method is an encapsulation of the task to be performed within the process unit whenever the processing resources are available. Typically, the server threads are the resources, which come from a hierarchical thread pool within the process unit. The threads within a thread pool are given an object of MSIQTask, with the MSIBasicQ node or an MSIHierarchicalQ node which they service. The method GetNextJob implements checking all subqs from a given level for a job. It returns timeout if there is no job after a scan of all basic q's under the given level.

```
void MSIQTask::Run(){
    ...
    // Code deleted: setup preferred path if any and
    // setup current weights for all subqueues with weighted share
    ...
    while(!IsCanceled()){
        if(mPreferredSubQ){
            IRC = GetNextPreferredJob(mPreferredSubQ, 0,
                                     mSubQIndexBelowFixed.begin(), &IJob, &IQ);
            IRC = mPreferredQ->WaitForNext(&IJob, gcMSIQMaxWaitTimeForNextJob);
            if(IRC==gcMSIQTimedOut){
                continue;
            }
            IRC = GetNextJob(mQ, 0, mFixedSubQIndex.begin(),
                             mNonPreferredSubQIndex.begin(), true, &IJob, &IQ);
            if(IRC == gcMSIQTimedOut){
                ITimedOut = true;
                continue;
            }
        }
        // else got job in a non preferred Q
    }
    // else got job in preferred Q
}
else{
    // no preferred Q
    IRC = GetNextJob(mQ, 0, mFixedSubQIndex.begin(),
                     mNonPreferredSubQIndex.begin(), true, &IJob, &IQ);
    if(IRC == gcMSIQTimedOut){
        ITimedOut = true;
        continue;
    }
}
...

mCheckGovernorsTask->SetJob(IJob);
mCheckGovernorsTask->Run();
if(mCheckGovernorsTask->GetStatus() == MSISJobTask::JOBTASK_ABORT);
...
}
else{
    ...
    IJob->AddRef(); // we need the job to be alive till we are done with it
    IJobId = IJob->GetId();
    IJob->SetPUStart(mOwnerid);
    try{
        mProcess->SetJob(IJob);
        mProcess->SetThread(GetThread());
        mProcess->Run();
    }
    catch(...){
        //MSIQTaskTrace_TEXT("ERROR: task execution for job threw an exception").
        IJobId, IQ);
    }
    mProcess->SetThread(NULL);
}
```

```

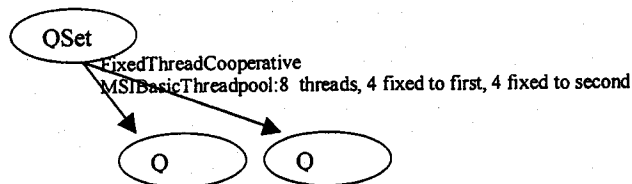
    }
    Job->SetPUExit(mOwnerId);
    IQStatus = IQ->Dequeue(Job);
    if(IQStatus == gcMSIQSucceeded) Job->Release();
    ThreadInfo.SetJobId(-1);
    ...
} // end of while
...
}

```

### 1.8.1 FixedThreadCooperative

When the servicing scheme is FixedThreadCooperative, the MSIQTask then also contains a PreferredSubQ path, which is the path down the hierarchy along which all nodes specify fixed thread cooperative. In this case, threads attempt to get a job from the preferred q/qset first and if failed, will attempt to get from any queue in the Qset. The method GetNextPreferredJob implements checking the preferred subq path for any job, which returns timeout if there is no job after one scan.

**Note:** This servicing scheme is meant to fix threads to subqs, but let them service other subq's at the same level only when there is no job in the subq to which the thread is fixed. Assume the following scenario.



A Qset contains two subqs, which are basic queues. The number of threads in the threadpools are eight, out of which four are fixed to first and the remaining four are fixed to the second. Suppose, at a particular instance the first subq had no jobs, and the four threads fixed to started processing jobs from the second subq which had a large number of jobs. Suppose a job arrives in the first subqueue during which time all four threads are still servicing the jobs they picked up from the second subq. At just such a time, one of the threads fixed to the second subq finishes one of the jobs and becomes available. Which subq will it pick the next job from?

There are two different interpretations of this servicing scheme which answer this question differently. The first interpretation is that the threads are fixed to the preferred subq they are servicing and as long as there are jobs in that subq, they do not attempt to take jobs from any other queue. According to this, the answer to the above question would be that the thread belonging to the second subq would take its next job from the second subq.

The second interpretation of this servicing scheme is that the threads are not fixed to the subq permanently, rather they are dynamically assigned to a subq's according to a fixed proportion given in the configuration. According to the above thread (any thread from the second subq that becomes available next), gets assigned to the first subq as soon as there is a job in the first subq. So, the threads would change their preferred queue to optimize in the situations described above. This makes more sense as automatic load balancing within a process unit.

In our current implementation, we have chosen to use the first interpretation and minimize the implementation complexity.

### 1.8.2 WeightedShare

When the servicing scheme is weighted share, the subqueues within the Qset also incorporate weights and a running count of number of jobs serviced at each subqueue. The method GetNextJob checks all subqs at the given level or a job and implements the running weight counter increments in a queue when returning a job from a that queue. The running counts are reset to zero when last subq count at any level reaches its maximum value.

### **1.8.3 HighestPriorityFirst**

The GetNextJob method implements checking all subq's at a given level or a job, in the order of highest priority to lowest priority subq.

**EXHIBIT B**

Scott's A3 Plan

ID	Task Name	Predecessors	% Work Complete	Work	Remaining Work	Start	Finish	Notes
1	<b>SERVER ADMIN</b>		92%	805.23 hrs	60.9 hrs	Tue 2/9/99	Thu 5/20/99	Owner: Scott
29	<b>Security: Application Access</b>		90%	91 hrs	9 hrs	Thu 3/11/99	Thu 5/20/99	to complete
39	<b>QE</b>		80%	20 hrs	4 hrs	Thu 3/25/99	Thu 5/20/99	
43	<b>Tests</b>		80%	20 hrs	4 hrs	Thu 4/1/99	Thu 5/20/99	
45	Total Time to create tests, run, and regress	38,44	80%	20 hrs	4 hrs	Thu 4/1/99	Thu 5/20/99	
49	<b>VLDB Properties</b>		94%	48 hrs	2.8 hrs	Tue 3/16/99	Thu 5/20/99	
56	<b>QE</b>		30%	4 hrs	2.8 hrs	Thu 4/1/99	Thu 5/20/99	
60	<b>Tests</b>		30%	4 hrs	2.8 hrs	Fri 5/7/99	Thu 5/20/99	
62	Total Time to create tests, run, and regress	55	30%	4 hrs	2.8 hrs	Thu 5/20/99	Thu 5/20/99	
65	<b>Caching: Admin and Monitoring</b>		88%	115 hrs	15.6 hrs	Tue 2/9/99	Thu 5/20/99	
79	<b>QE</b>		45%	12 hrs	6.6 hrs	Thu 4/15/99	Thu 5/20/99	
83	<b>Tests</b>		45%	12 hrs	6.6 hrs	Wed 5/5/99	Thu 5/20/99	
85	Total Time to create tests, run, and regress	84	45%	12 hrs	6.6 hrs	Wed 5/19/99	Thu 5/20/99	
89	<b>Project Configuration</b>		84%	56 hrs	8.9 hrs	Thu 2/18/99	Thu 5/20/99	
96	<b>QE</b>		65%	14 hrs	4.9 hrs	Wed 4/14/99	Thu 5/20/99	
100	<b>Tests</b>		65%	14 hrs	4.9 hrs	Fri 4/16/99	Thu 5/20/99	
102	Total Time to create tests, run, and regress		65%	14 hrs	4.9 hrs	Wed 5/19/99	Thu 5/20/99	
117	<b>Cluster Admin</b>		93%	45 hrs	3 hrs	Wed 3/17/99	Thu 5/20/99	
125	<b>QE</b>		80%	10 hrs	2 hrs	Thu 4/1/99	Thu 5/20/99	
129	<b>Tests</b>		80%	10 hrs	2 hrs	Tue 4/20/99	Thu 5/20/99	
131	Total Time to create tests, run, and regress	121,127	80%	10 hrs	2 hrs	Wed 5/19/99	Thu 5/20/99	
144	<b>Database Objects</b>		93%	124.02 hrs	9.2 hrs	Tue 3/9/99	Thu 5/20/99	
156	<b>QE</b>		80%	26 hrs	5.2 hrs	Tue 4/27/99	Thu 5/20/99	
160	<b>Tests</b>		80%	26 hrs	5.2 hrs	Tue 5/4/99	Thu 5/20/99	
162	Total Time to create tests, run, and regress	154,161	80%	26 hrs	5.2 hrs	Fri 5/14/99	Thu 5/20/99	
176	<b>SCHEDULING</b>		89%	156 hrs	16.6 hrs	Tue 2/9/99	Fri 5/21/99	Owner: Scott
177	<b>Scheduling: General</b>		82%	71 hrs	12.6 hrs	Fri 3/5/99	Fri 5/21/99	
178	<b>QE</b>		83%	70 hrs	11.6 hrs	Fri 3/5/99	Fri 5/21/99	
184	<b>Tests</b>		80%	58 hrs	11.6 hrs	Tue 3/23/99	Fri 5/21/99	
186	Total Time to create tests, run, and regress	185	80%	30 hrs	6 hrs	Tue 4/6/99	Fri 5/21/99	
187	Feature testing - back end	210	80%	28 hrs	5.6 hrs	Tue 3/23/99	Wed 4/14/99	
213	<b>JOB PRIORITIZATION AND SERVICING</b>		98%	213 hrs	5 hrs	Tue 2/9/99	Fri 5/21/99	Owner: Scott

Scott's A3 Plan

ID	Task Name	Predecessors	% Work Complete	Work	Remaining Work	Start	Finish	Notes
214	<b>Job Prioritization</b>		<b>98%</b>	<b>213 hrs</b>	<b>5 hrs</b>	<b>Tue 2/9/99</b>	<b>Fri 5/21/99</b>	
241	<b>QE</b>		<b>88%</b>	<b>28 hrs</b>	<b>4 hrs</b>	<b>Tue 2/9/99</b>	<b>Fri 5/21/99</b>	
247	<b>Tests</b>		<b>75%</b>	<b>16 hrs</b>	<b>4 hrs</b>	<b>Wed 3/10/99</b>	<b>Fri 5/21/99</b>	
249	Feature testing - front end	228,243	70%	8 hrs	2.4 hrs	Thu 5/20/99	Fri 5/21/99	
250	Feature testing - back end	228,245	80%	8 hrs	1.6 hrs	Wed 3/10/99	Wed 3/10/99	
254	<b>CONFIGURATION WIZARD</b>		<b>86%</b>	<b>206 hrs</b>	<b>28.4 hrs</b>	<b>Tue 2/9/99</b>	<b>Thu 5/20/99</b>	<b>Owner: Scott</b>
304	<b>Control Panel Applet</b>		<b>42%</b>	<b>37 hrs</b>	<b>21.4 hrs</b>	<b>Fri 3/19/99</b>	<b>Fri 5/14/99</b>	
312	<b>QE</b>		<b>20%</b>	<b>8 hrs</b>	<b>6.4 hrs</b>	<b>Tue 5/4/99</b>	<b>Fri 5/14/99</b>	
316	<b>Tests</b>		<b>20%</b>	<b>8 hrs</b>	<b>6.4 hrs</b>	<b>Tue 5/4/99</b>	<b>Fri 5/14/99</b>	
318	Total Time to create tests, run, and regress	309,317	20%	8 hrs	6.4 hrs	Fri 5/14/99	Fri 5/14/99	



Scott's A3 Plan

Resource Names	7, '99																												Feb 14, '99							Feb 21, '99							Feb 28, '99							Mar 7, '99						
	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F																							
Chao Frances - QE																																																								
Weld David - QE																																																								
Marchal Olivier - QE																																																								

Weld D



Scott's A3 Plan

Mar 14, '99							Mar 21, '99							Mar 28, '99							Apr 4, '99							Apr 11, '99							Apr 18, '99		
S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M

Handwritten notes and arrows on the A3 Plan grid:

- Top section (Mar 14 - Apr 18):
  - Arrow from Mar 21 to Mar 28.
  - Arrow from Mar 28 to Apr 4.
  - Arrow from Apr 4 to Apr 11.
  - Arrow from Apr 11 to Apr 18.
- Middle section (Mar 21 - Apr 18):
  - Arrow from Mar 21 to Mar 28.
  - Arrow from Mar 28 to Apr 4.
  - Arrow from Apr 4 to Apr 11.
  - Arrow from Apr 11 to Apr 18.
- Bottom section (Mar 21 - Apr 18):
  - Arrow from Mar 21 to Mar 28.
  - Arrow from Mar 28 to Apr 4.
  - Arrow from Apr 4 to Apr 11.
  - Arrow from Apr 11 to Apr 18.

Weld David - QE





May 23, '99			
C	M	T	

T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

**Marchal Olivier - QE**

ID	Task Name	% Complete	Work	Start	Finish	redecessor	Resource Names
1	<b>Datamarting (Design only)</b>	100%	32 hrs	Tue 2/9/99	Fri 3/19/99		
2	<b>Design</b>	100%	32 hrs	Tue 2/9/99	Fri 3/19/99		
3	Design Draft (metric alias)	100%	8 hrs	Tue 2/9/99	Wed 2/10/99		Li Benjamin Z. - DES
4	Design Review & Update	100%	16 hrs	Thu 2/11/99	Fri 2/12/99	3	Li Benjamin Z. - DES, Yuan Jun - DES
5	Engineer Review	100%	8 hrs	Thu 3/18/99	Fri 3/19/99	4	Liao Rixin - DES
6	<b>Null/Zero Handling</b>	36%	72 hrs	Tue 3/23/99	Wed 3/31/99	1	
7	<b>Design</b>	100%	18 hrs	Tue 3/23/99	Wed 3/24/99		
8	Design Draft	100%	16 hrs	Tue 3/23/99	Tue 3/23/99		Yuan Jun - DES, Li Benjamin Z. - DES, Liao Rixin - DES
9	Design Review & Update	100%	2 hrs	Wed 3/24/99	Wed 3/24/99	8	Benjamin Z. - DES, Wang Xinyi - DES, Liao Rixin - DES
10	<b>Implementation</b>	100%	16 hrs	Wed 3/24/99	Thu 3/25/99	7	
11	Null/Zero Function	100%	4 hrs	Wed 3/24/99	Wed 3/24/99		Ma Yuling - IMP
12	SQL Generation and Population Instruction	100%	12 hrs	Wed 3/24/99	Thu 3/25/99		Liao Rixin - IMP
13	COM dependency ?	100%	0 hrs	Wed 3/24/99	Wed 3/24/99		COM - IMP
14	<b>Standard work</b>	0%	5 hrs	Thu 3/25/99	Thu 3/25/99	10	
15	Memory leak, Diagnostics work	0%	4 hrs	Thu 3/25/99	Thu 3/25/99		Ma Yuling - SW, Liao Rixin - SW
16	Code Review	0%	1 hr	Thu 3/25/99	Thu 3/25/99		Yuan Jun - SW, Liao Rixin - SW
17	<b>Integration</b>	8%	5 hrs	Fri 3/26/99	Fri 3/26/99	14	
18	Backend Integration	50%	1 hr	Fri 3/26/99	Fri 3/26/99		Ma Yuling - INT, Liao Rixin - INT
19	End to end integration review	0%	1 hr	Fri 3/26/99	Fri 3/26/99	11	Paz Andres - INT
20	End to End Integration	0%	3 hrs	Fri 3/26/99	Fri 3/26/99	18	Ma Yuling - INT, Liao Rixin - INT
21	<b>QE</b>	0%	28 hrs	Wed 3/24/99	Wed 3/31/99		
22	<b>Test Plan</b>	0%	8 hrs	Wed 3/24/99	Thu 3/25/99	7	
23	QE Test Draft	0%	4 hrs	Wed 3/24/99	Wed 3/24/99		Busse Stephane - QE
24	QE Test Review & Update	0%	4 hrs	Wed 3/24/99	Thu 3/25/99	23	Busse Stephane - QE
25	<b>Tests</b>	0%	20 hrs	Fri 3/26/99	Wed 3/31/99	17,22	
26	New Feature Testing	0%	16 hrs	Fri 3/26/99	Wed 3/31/99		Busse Stephane - QE
27	Incorporation into Regression	0%	4 hrs	Wed 3/31/99	Wed 3/31/99	26	Busse Stephane - QE
28	<b>Governing</b>	0%	78 hrs	Mon 3/29/99	Wed 4/14/99	17	
29	<b>Functional Specification</b>	0%	0 hrs	Mon 3/29/99	Mon 3/29/99		
30	Spec draft	100%	0 hrs	Mon 3/29/99	Mon 3/29/99		Cappiello Scott - SPEC, Everhart L. Douglas - SPEC
31	Spec Review & Update	0%	0 hrs	Mon 3/29/99	Mon 3/29/99	30	Cappiello Scott - SPEC
32	<b>Design</b>	0%	20 hrs	Tue 3/30/99	Tue 4/6/99	29	
33	Design Draft	0%	12 hrs	Tue 3/30/99	Thu 4/1/99		Liao Rixin - DES
34	Design Review & Update	0%	8 hrs	Fri 4/2/99	Tue 4/6/99	33	Yuan Jun - DES
35	<b>Implementation</b>	0%	28 hrs	Tue 4/6/99	Thu 4/8/99	32	
36	DB Class Enhancement	0%	0 hrs	Tue 4/6/99	Tue 4/6/99		Dixit Sunil - IMP

ID	Task Name	% Complete	Work	Start	Finish	redecessor	Resource Names
37	Kernel dependency (need govern info)	0%	0 hrs	Tue 4/6/99	Wed 4/7/99	36	
38	Incremental Fetching	0%	12 hrs	Wed 4/7/99	Thu 4/8/99	37	Liao Rixin - IMP
39	Maximal Row Return	0%	4 hrs	Tue 4/6/99	Tue 4/6/99	36	Liao Rixin - IMP
40	Timeout Handling	0%	4 hrs	Tue 4/6/99	Tue 4/6/99	34	Liao Rixin - IMP
41	Other Implementation	0%	8 hrs	Tue 4/6/99	Wed 4/7/99	34	Liao Rixin - IMP
42	<b>Standard work</b>	<b>0%</b>	<b>8 hrs</b>	<b>Fri 4/9/99</b>	<b>Fri 4/9/99</b>	<b>35</b>	
43	0 Memory Leak Certification	0%	6 hrs	Fri 4/9/99	Fri 4/9/99		Liao Rixin - SW
44	Code Review	0%	2 hrs	Fri 4/9/99	Fri 4/9/99	43	Yuan Jun - SW,Liao Rixin - SW
45	<b>Integration</b>	<b>0%</b>	<b>2 hrs</b>	<b>Tue 4/13/99</b>	<b>Tue 4/13/99</b>	<b>42</b>	
46	Backend Integration	0%	1 hr	Tue 4/13/99	Tue 4/13/99		Liao Rixin - INT
47	End to End Integration	0%	1 hr	Tue 4/13/99	Tue 4/13/99	46	Liao Rixin - INT
48	<b>QE</b>	<b>0%</b>	<b>20 hrs</b>	<b>Tue 4/6/99</b>	<b>Wed 4/14/99</b>		
49	<b>Test Plan</b>	<b>0%</b>	<b>8 hrs</b>	<b>Tue 4/6/99</b>	<b>Wed 4/7/99</b>	<b>32</b>	
50	QE Test Draft	0%	4 hrs	Tue 4/6/99	Tue 4/6/99		Yang An (Andre) - QE
51	QE Test Review & Update	0%	4 hrs	Wed 4/7/99	Wed 4/7/99	50	Yang An (Andre) - QE
52	<b>Tests</b>	<b>0%</b>	<b>12 hrs</b>	<b>Tue 4/13/99</b>	<b>Wed 4/14/99</b>	<b>45,49</b>	
53	New Feature Testing	0%	6 hrs	Tue 4/13/99	Tue 4/13/99		Yang An (Andre) - QE
54	Incorporation into Regression	0%	6 hrs	Wed 4/14/99	Wed 4/14/99	53	Yang An (Andre) - QE
55	<b>Count/Rank Consider NULL</b>	<b>0%</b>	<b>90 hrs</b>	<b>Tue 4/13/99</b>	<b>Wed 4/21/99</b>	<b>45</b>	
56	<b>Design</b>	<b>0%</b>	<b>25 hrs</b>	<b>Tue 4/13/99</b>	<b>Thu 4/15/99</b>		
57	Design Draft	0%	18 hrs	Tue 4/13/99	Wed 4/14/99		IES,Yuan Jun - DES,Ma Yuling - DES,Liao Rixin - DES
58	Design Review & Update	0%	7 hrs	Wed 4/14/99	Thu 4/15/99	57	IES,Yuan Jun - DES,Ma Yuling - DES,Liao Rixin - DES
59	<b>Implementation</b>	<b>0%</b>	<b>32 hrs</b>	<b>Thu 4/15/99</b>	<b>Fri 4/16/99</b>	<b>56</b>	
60	Measure/Filter Editor	0%	4 hrs	Thu 4/15/99	Thu 4/15/99		Paz Andres - IMP
61	VLDB Syntax	0%	8 hrs	Thu 4/15/99	Fri 4/16/99		Chen Yirong - IMP
62	Filter/Metric Process	0%	20 hrs	Thu 4/15/99	Fri 4/16/99		Ma Yuling - IMP,Liao Rixin - IMP
63	Function Server Support	0%	0 hrs	Thu 4/15/99	Thu 4/15/99		Ma Yuling - IMP
64	<b>Standard work</b>	<b>0%</b>	<b>7 hrs</b>	<b>Fri 4/16/99</b>	<b>Tue 4/20/99</b>	<b>59</b>	
65	Memory Leak, Diagnostics work	0%	5 hrs	Fri 4/16/99	Fri 4/16/99		Liao Rixin - SW
66	Code Review	0%	2 hrs	Fri 4/16/99	Tue 4/20/99	65	Yuan Jun - SW,Liao Rixin - SW
67	<b>Integration</b>	<b>0%</b>	<b>2 hrs</b>	<b>Tue 4/20/99</b>	<b>Tue 4/20/99</b>	<b>64</b>	
68	End to End Integration	0%	2 hrs	Tue 4/20/99	Tue 4/20/99		Yuan Jun - INT,Ma Yuling - INT
69	<b>QE</b>	<b>0%</b>	<b>24 hrs</b>	<b>Thu 4/15/99</b>	<b>Wed 4/21/99</b>		
70	<b>Test Plan</b>	<b>0%</b>	<b>8 hrs</b>	<b>Thu 4/15/99</b>	<b>Fri 4/16/99</b>	<b>56</b>	
71	QE Test Draft	0%	4 hrs	Thu 4/15/99	Thu 4/15/99		Busse Stephane - QE
72	QE Test Review & Update	0%	4 hrs	Thu 4/15/99	Fri 4/16/99	71	Busse Stephane - QE

ID	Task Name	% Complete	Work	Start	Finish	redecessor	Resource Names
73	<b>Tests</b>	<b>0%</b>	<b>16 hrs</b>	<b>Tue 4/20/99</b>	<b>Wed 4/21/99</b>	<b>67,70</b>	
74	New Feature Testing	0%	12 hrs	Tue 4/20/99	Tue 4/20/99		Busse Stephane - QE,Cheng Lingxiang - QE
75	Incorporation into Regression	0%	4 hrs	Tue 4/20/99	Wed 4/21/99	74	Busse Stephane - QE
76	<b>Non-agg w/ transformation</b>	<b>45%</b>	<b>81 hrs</b>	<b>Tue 3/30/99</b>	<b>Wed 4/14/99</b>		
77	<b>Design</b>	<b>100%</b>	<b>12 hrs</b>	<b>Tue 3/30/99</b>	<b>Thu 4/1/99</b>		
78	Design Draft	100%	8 hrs	Tue 3/30/99	Wed 3/31/99		Yuan Jun - DES
79	Design Review & Update	100%	4 hrs	Thu 4/1/99	Thu 4/1/99	78	Li Benjamin Z. - DES
80	<b>Implementation</b>	<b>100%</b>	<b>24 hrs</b>	<b>Thu 4/1/99</b>	<b>Wed 4/7/99</b>	<b>77</b>	
81	Implementation	100%	24 hrs	Thu 4/1/99	Wed 4/7/99		Yuan Jun - IMP
82	<b>Standard work</b>	<b>0%</b>	<b>25 hrs</b>	<b>Wed 4/7/99</b>	<b>Fri 4/9/99</b>	<b>80</b>	
83	Regression testing work	0%	19 hrs	Wed 4/7/99	Fri 4/9/99		Yuan Jun - SW
84	0 Memory Leak Certification	0%	4 hrs	Wed 4/7/99	Wed 4/7/99		Yuan Jun - SW
85	Code Review	0%	2 hrs	Fri 4/9/99	Fri 4/9/99	84,83	Yuan Jun - SW, Li Benjamin Z. - SW
86	<b>QE</b>	<b>0%</b>	<b>20 hrs</b>	<b>Thu 4/1/99</b>	<b>Wed 4/14/99</b>		
87	<b>Test Plan</b>	<b>0%</b>	<b>8 hrs</b>	<b>Thu 4/1/99</b>	<b>Fri 4/2/99</b>	<b>77</b>	
88	QE Test Draft	0%	4 hrs	Thu 4/1/99	Thu 4/1/99		Busse Stephane - QE
89	QE Test Review & Update	0%	4 hrs	Fri 4/2/99	Fri 4/2/99	88	Busse Stephane - QE
90	<b>Tests</b>	<b>0%</b>	<b>12 hrs</b>	<b>Tue 4/13/99</b>	<b>Wed 4/14/99</b>	<b>82,87</b>	
91	New Feature Testing	0%	8 hrs	Tue 4/13/99	Tue 4/13/99		Busse Stephane - QE
92	Incorporation into Regression	0%	4 hrs	Wed 4/14/99	Wed 4/14/99	91	Busse Stephane - QE
93	<b>Nested Agregation (SQL engine only)</b>	<b>0%</b>	<b>75 hrs</b>	<b>Tue 4/13/99</b>	<b>Tue 4/20/99</b>	<b>82</b>	
94	<b>Design</b>	<b>0%</b>	<b>4 hrs</b>	<b>Tue 4/13/99</b>	<b>Tue 4/13/99</b>		
95	Design Review	0%	4 hrs	Tue 4/13/99	Tue 4/13/99		Yuan Jun - DES,Wang Xinyi - DES
96	<b>Implementation</b>	<b>0%</b>	<b>12 hrs</b>	<b>Tue 4/13/99</b>	<b>Wed 4/14/99</b>	<b>94</b>	
97	Enhanced Filtering Rule	0%	10 hrs	Tue 4/13/99	Wed 4/14/99		Yuan Jun - IMP
98	Analytical Engine Support	0%	2 hrs	Tue 4/13/99	Tue 4/13/99		Wang Xinyi - IMP
99	<b>Standard work</b>	<b>0%</b>	<b>5 hrs</b>	<b>Wed 4/14/99</b>	<b>Thu 4/15/99</b>	<b>96</b>	
100	Memory Leak, Diagnostics work	0%	4 hrs	Wed 4/14/99	Wed 4/14/99		Yuan Jun - SW
101	Code Review	0%	1 hr	Thu 4/15/99	Thu 4/15/99	100	Yuan Jun - SW
102	<b>Integration</b>	<b>0%</b>	<b>2 hrs</b>	<b>Thu 4/15/99</b>	<b>Thu 4/15/99</b>	<b>99</b>	
103	Backend Integration	0%	2 hrs	Thu 4/15/99	Thu 4/15/99		Yuan Jun - INT,Wang Xinyi - INT
104	<b>QE</b>	<b>0%</b>	<b>52 hrs</b>	<b>Tue 4/13/99</b>	<b>Tue 4/20/99</b>		
105	<b>Test Plan</b>	<b>0%</b>	<b>24 hrs</b>	<b>Tue 4/13/99</b>	<b>Wed 4/14/99</b>	<b>94</b>	
106	QE Test Draft	0%	16 hrs	Tue 4/13/99	Wed 4/14/99		Busse Stephane - QE,Cheng Lingxiang - QE
107	QE Test Review & Update	0%	8 hrs	Wed 4/14/99	Wed 4/14/99	106	Busse Stephane - QE,Cheng Lingxiang - QE
108	<b>Tests</b>	<b>0%</b>	<b>28 hrs</b>	<b>Thu 4/15/99</b>	<b>Tue 4/20/99</b>	<b>102,105</b>	

ID	Task Name	% Complete	Work	Start	Finish	Predecessor	Resource Names
109	New Feature Testing	0%	24 hrs	Thu 4/15/99	Fri 4/16/99		Busse Stephane - QE,Cheng Lingxiang - QE
110	Incorporation into Regression	0%	4 hrs	Fri 4/16/99	Tue 4/20/99	109	Busse Stephane - QE
111	<b>Analytical Function on Fact</b>	<b>0%</b>	<b>41 hrs</b>	<b>Thu 4/15/99</b>	<b>Thu 4/22/99</b>	<b>102</b>	
112	<b>Design</b>	<b>0%</b>	<b>6 hrs</b>	<b>Thu 4/15/99</b>	<b>Fri 4/16/99</b>		
113	Design Draft	0%	4 hrs	Thu 4/15/99	Thu 4/15/99		Wang Xinyi - DES,Yuan Jun - DES
114	Design Review & Update	0%	2 hrs	Fri 4/16/99	Fri 4/16/99	113	Wang Xinyi - DES,Yuan Jun - DES
115	<b>Implementation</b>	<b>0%</b>	<b>8 hrs</b>	<b>Fri 4/16/99</b>	<b>Tue 4/20/99</b>	<b>112</b>	
116	SQL Engine	0%	8 hrs	Fri 4/16/99	Tue 4/20/99		Yuan Jun - IMP
117	Analytical Engine	0%	0 hrs	Fri 4/16/99	Fri 4/16/99		Wang Xinyi - IMP
118	<b>Standard work</b>	<b>0%</b>	<b>5 hrs</b>	<b>Tue 4/20/99</b>	<b>Tue 4/20/99</b>	<b>115</b>	
119	Memory Leak, Diagnostics work	0%	3 hrs	Tue 4/20/99	Tue 4/20/99		Wang Xinyi - SW,Yuan Jun - SW
120	Code Review	0%	2 hrs	Tue 4/20/99	Tue 4/20/99	119	Li Benjamin Z. - SW
121	<b>Integration</b>	<b>0%</b>	<b>2 hrs</b>	<b>Wed 4/21/99</b>	<b>Wed 4/21/99</b>	<b>118</b>	
122	End to End Integration	0%	2 hrs	Wed 4/21/99	Wed 4/21/99		Wang Xinyi - INT,Yuan Jun - INT
123	<b>QE</b>	<b>0%</b>	<b>20 hrs</b>	<b>Fri 4/16/99</b>	<b>Thu 4/22/99</b>		
124	<b>Test Plan</b>	<b>0%</b>	<b>8 hrs</b>	<b>Fri 4/16/99</b>	<b>Tue 4/20/99</b>	<b>112</b>	
125	QE Test Draft	0%	4 hrs	Fri 4/16/99	Fri 4/16/99		Busse Stephane - QE
126	QE Test Review & Update	0%	4 hrs	Fri 4/16/99	Tue 4/20/99	125	Busse Stephane - QE
127	<b>Tests</b>	<b>0%</b>	<b>12 hrs</b>	<b>Wed 4/21/99</b>	<b>Thu 4/22/99</b>	<b>121,124</b>	
128	New Feature Testing	0%	8 hrs	Wed 4/21/99	Thu 4/22/99		Busse Stephane - QE
129	Incorporation into Regression	0%	4 hrs	Thu 4/22/99	Thu 4/22/99	128	Busse Stephane - QE
130	<b>OLAP Function of RDBMS</b>	<b>5%</b>	<b>48 hrs</b>	<b>Wed 4/21/99</b>	<b>Thu 4/29/99</b>	<b>121</b>	
131	<b>Specification</b>	<b>0%</b>	<b>0 hrs</b>	<b>Wed 4/21/99</b>	<b>Wed 4/21/99</b>		
132	DBMS Research	85%	0 hrs	Wed 4/21/99	Wed 4/21/99	170	Engine - SPEC
133	Specification Draft	85%	0 hrs	Wed 4/21/99	Wed 4/21/99	132	Bedell Jeffrey - SPEC
134	<b>Design</b>	<b>27%</b>	<b>10 hrs</b>	<b>Wed 4/21/99</b>	<b>Fri 4/23/99</b>	<b>131</b>	
135	Design Draft	30%	8 hrs	Wed 4/21/99	Fri 4/23/99		Yuan Jun - DES
136	Design Review & Update	0%	2 hrs	Fri 4/23/99	Fri 4/23/99	135	Yuan Jun - DES, Li Benjamin Z. - DES
137	<b>Implementation</b>	<b>0%</b>	<b>12 hrs</b>	<b>Fri 4/23/99</b>	<b>Fri 4/23/99</b>	<b>134</b>	
138	Implementation	0%	4 hrs	Fri 4/23/99	Fri 4/23/99		Yuan Jun - IMP
139	Integration with Function Server	0%	4 hrs	Fri 4/23/99	Fri 4/23/99		Ma Yuling - IMP
140	Pattern from VLDB	0%	4 hrs	Fri 4/23/99	Fri 4/23/99		Chen Yinong - IMP
141	<b>Standard work</b>	<b>0%</b>	<b>4 hrs</b>	<b>Wed 4/28/99</b>	<b>Wed 4/28/99</b>	<b>137</b>	
142	0 Memory Leak Certification	0%	2 hrs	Wed 4/28/99	Wed 4/28/99		Yuan Jun - SW
143	Code Review	0%	2 hrs	Wed 4/28/99	Wed 4/28/99	142	Yuan Jun - SW
144	<b>Integration</b>	<b>0%</b>	<b>4 hrs</b>	<b>Wed 4/28/99</b>	<b>Wed 4/28/99</b>	<b>141</b>	



ID	Task Name	% Complete	Work	Start	Finish	redecessor	Resource Names
145	Backend Integration	0%	2 hrs	Wed 4/28/99	Wed 4/28/99		Yuan Jun - INT
146	End to End Integration	0%	2 hrs	Wed 4/28/99	Wed 4/28/99	145	Yuan Jun - INT
147	<b>QE</b>	<b>0%</b>	<b>18 hrs</b>	<b>Fri 4/23/99</b>	<b>Thu 4/29/99</b>		
148	<b>Test Plan</b>	<b>0%</b>	<b>10 hrs</b>	<b>Fri 4/23/99</b>	<b>Wed 4/28/99</b>	<b>134</b>	
149	QE Test Draft	0%	6 hrs	Fri 4/23/99	Wed 4/28/99		Busse Stephane - QE
150	QE Test Review & Update	0%	4 hrs	Wed 4/28/99	Wed 4/28/99	149	Busse Stephane - QE
151	<b>Tests</b>	<b>0%</b>	<b>8 hrs</b>	<b>Wed 4/28/99</b>	<b>Thu 4/29/99</b>	<b>144,148</b>	
152	New Feature Testing	0%	6 hrs	Wed 4/28/99	Thu 4/29/99		Busse Stephane - QE
153	Incorporation into Regression	0%	2 hrs	Thu 4/29/99	Thu 4/29/99	152	Busse Stephane - QE
154	<b>Smart Totalling</b>	<b>0%</b>	<b>130 hrs</b>	<b>Tue 4/6/99</b>	<b>Wed 4/21/99</b>		
155	<b>Design</b>	<b>0%</b>	<b>20 hrs</b>	<b>Tue 4/6/99</b>	<b>Wed 4/7/99</b>		
156	Design Draft	0%	16 hrs	Tue 4/6/99	Tue 4/6/99		Yuan Jun - DES,Feng Xun - DES,LI Benjamin Z. - DES
157	Design Review & Update	0%	4 hrs	Wed 4/7/99	Wed 4/7/99	156	Jan Jun - DES,LI Benjamin Z. - DES,Wang Xinyi - DES
158	<b>Implementation</b>	<b>0%</b>	<b>68 hrs</b>	<b>Wed 4/7/99</b>	<b>Wed 4/14/99</b>	<b>155</b>	
159	DFC Conversion	0%	16 hrs	Wed 4/7/99	Fri 4/9/99		Feng Xun - IMP
160	SQL Generation and Population Instruction	0%	28 hrs	Wed 4/7/99	Tue 4/13/99	157	Feng Xun - IMP
161	Data Population	0%	16 hrs	Tue 4/13/99	Wed 4/14/99	160	Feng Xun - IMP,Wang Xinyi - IMP
162	GUI dependency ?	0%	0 hrs	Wed 4/7/99	Wed 4/7/99		Paz Andres - IMP
163	COM dependency ?	0%	0 hrs	Wed 4/7/99	Wed 4/7/99		COM - IMP
164	Smart Total	0%	8 hrs	Wed 4/7/99	Thu 4/8/99		Ma Yuling - IMP
165	<b>Standard work</b>	<b>0%</b>	<b>8 hrs</b>	<b>Wed 4/14/99</b>	<b>Thu 4/15/99</b>	<b>158</b>	
166	Memory leak, Diagnostics work	0%	6 hrs	Wed 4/14/99	Wed 4/14/99		Feng Xun - SW,Wang Xinyi - SW,Ma Yuling - SW
167	Code Review	0%	2 hrs	Wed 4/14/99	Thu 4/15/99	166	Yuan Jun - SW
168	<b>Integration</b>	<b>0%</b>	<b>6 hrs</b>	<b>Thu 4/15/99</b>	<b>Fri 4/16/99</b>	<b>165</b>	
169	Backend Integration	0%	4 hrs	Thu 4/15/99	Fri 4/16/99		Yuan Jun - INT,Yuan Jun - INT,Feng Xun - INT,Ma Yuling - INT
170	End to End Integration	0%	2 hrs	Fri 4/16/99	Fri 4/16/99	169	Yuan Jun - INT,Yuan Jun - INT,Feng Xun - INT,Ma Yuling - INT
171	<b>QE</b>	<b>0%</b>	<b>28 hrs</b>	<b>Wed 4/7/99</b>	<b>Wed 4/21/99</b>		
172	<b>Test Plan</b>	<b>0%</b>	<b>8 hrs</b>	<b>Wed 4/7/99</b>	<b>Thu 4/8/99</b>	<b>155</b>	
173	QE Test Draft	0%	5 hrs	Wed 4/7/99	Wed 4/7/99		Busse Stephane - QE
174	QE Test Review & Update	0%	3 hrs	Wed 4/7/99	Thu 4/8/99	173	Busse Stephane - QE
175	<b>Tests</b>	<b>0%</b>	<b>20 hrs</b>	<b>Fri 4/16/99</b>	<b>Wed 4/21/99</b>	<b>168,172</b>	
176	New Feature Testing	0%	16 hrs	Fri 4/16/99	Wed 4/21/99		Busse Stephane - QE
177	Incorporation into Regression	0%	4 hrs	Wed 4/21/99	Wed 4/21/99	176	Busse Stephane - QE
178	<b>SQL Cancel</b>	<b>0%</b>	<b>63 hrs</b>	<b>Fri 4/16/99</b>	<b>Thu 4/29/99</b>	<b>168</b>	
179	<b>Functional Specification</b>	<b>0%</b>	<b>0 hrs</b>	<b>Fri 4/16/99</b>	<b>Fri 4/16/99</b>		
180	Spec Draft	0%	0 hrs	Fri 4/16/99	Fri 4/16/99		Cappiello Scott - SPEC

ID	Task Name	% Complete	Work	Start	Finish	redecessor	Resource Names
181	Spec Review & Update	0%	0 hrs	Fri 4/16/99	Fri 4/16/99	180	Cappiello Scott - SPEC
182	<b>Design</b>	<b>0%</b>	<b>8 hrs</b>	<b>Fri 4/16/99</b>	<b>Tue 4/20/99</b>	<b>179</b>	
183	ODBC Research	0%	0 hrs	Fri 4/16/99	Fri 4/16/99		
184	Design Draft	0%	6 hrs	Fri 4/16/99	Fri 4/16/99		Dixit Sunil - DES
185	Design Review & Update	0%	2 hrs	Fri 4/16/99	Tue 4/20/99	184	
186	<b>Implementation</b>	<b>0%</b>	<b>14 hrs</b>	<b>Tue 4/20/99</b>	<b>Wed 4/21/99</b>	<b>182</b>	Li Benjamin Z. - DES
187	Implementation	0%	10 hrs	Tue 4/20/99	Wed 4/21/99		Feng Xun - IMP
188	Kernel Call	0%	4 hrs	Tue 4/20/99	Tue 4/20/99		Polana Ramprasad - IMP
189	<b>Standard work</b>	<b>0%</b>	<b>5 hrs</b>	<b>Wed 4/21/99</b>	<b>Thu 4/22/99</b>	<b>186</b>	
190	0 Memory Leak Certification	0%	4 hrs	Wed 4/21/99	Wed 4/21/99		Feng Xun - SW
191	Code Review	0%	1 hr	Wed 4/21/99	Thu 4/22/99	190	Yuan Jun - SW
192	<b>Integration</b>	<b>0%</b>	<b>2 hrs</b>	<b>Thu 4/22/99</b>	<b>Thu 4/22/99</b>	<b>189</b>	Polana Ramprasad - INT, Liao Rixin - INT
193	End to End Integration	0%	2 hrs	Thu 4/22/99	Thu 4/22/99		
194	<b>QE</b>	<b>0%</b>	<b>34 hrs</b>	<b>Tue 4/20/99</b>	<b>Thu 4/23/99</b>		
195	<b>Test Plan</b>	<b>0%</b>	<b>10 hrs</b>	<b>Tue 4/20/99</b>	<b>Wed 4/21/99</b>	<b>182</b>	
196	QE Test Draft	0%	6 hrs	Tue 4/20/99	Tue 4/20/99		Yang An (Andre) - QE
197	QE Test Review & Update	0%	4 hrs	Tue 4/20/99	Wed 4/21/99	196	Yang An (Andre) - QE
198	<b>Tests</b>	<b>0%</b>	<b>24 hrs</b>	<b>Thu 4/22/99</b>	<b>Thu 4/23/99</b>	<b>192, 195</b>	
199	Unit Testing	0%	4 hrs	Thu 4/22/99	Thu 4/22/99		Yang An (Andre) - QE
200	New Feature Testing	0%	16 hrs	Thu 4/22/99	Wed 4/28/99	199	Wang Jianhua - QE
201	Incorporation into Regression	0%	4 hrs	Wed 4/28/99	Thu 4/29/99	200	Yang An (Andre) - QE
202	<b>Catalog Look Workarounds</b>	<b>2%</b>	<b>34 hrs</b>	<b>Thu 4/22/99</b>	<b>Tue 5/4/99</b>	<b>192</b>	
203	<b>Functional Specification</b>	<b>100%</b>	<b>0 hrs</b>	<b>Thu 4/22/99</b>	<b>Thu 4/22/99</b>		
204	Spec draft	100%	0 hrs	Thu 4/22/99	Thu 4/22/99		Bedell Jeffrey - SPEC
205	Spec Review & Update	100%	0 hrs	Thu 4/22/99	Thu 4/22/99	204	Bedell Jeffrey - SPEC
206	<b>Design</b>	<b>15%</b>	<b>4 hrs</b>	<b>Thu 4/22/99</b>	<b>Fri 4/23/99</b>	<b>203</b>	
207	Design Draft	30%	2 hrs	Thu 4/22/99	Thu 4/22/99		Yuan Jun - DES
208	Design Review & Update	0%	2 hrs	Thu 4/22/99	Fri 4/23/99	207	Yuan Jun - DES
209	<b>Implementation</b>	<b>0%</b>	<b>8 hrs</b>	<b>Wed 4/28/99</b>	<b>Wed 4/28/99</b>	<b>206</b>	
210	Tasks to be added once Spec and Design are (	0%	8 hrs	Wed 4/28/99	Wed 4/28/99		Feng Xun - IMP
211	<b>Standard work</b>	<b>0%</b>	<b>2 hrs</b>	<b>Thu 4/29/99</b>	<b>Thu 4/29/99</b>	<b>209</b>	
212	0 Memory Leak Certification	0%	1 hr	Thu 4/29/99	Thu 4/29/99		Feng Xun - SW
213	Code Review	0%	1 hr	Thu 4/29/99	Thu 4/29/99	212	Yuan Jun - SW
214	<b>Integration</b>	<b>0%</b>	<b>2 hrs</b>	<b>Fri 4/30/99</b>	<b>Fri 4/30/99</b>	<b>211</b>	
215	End to End Integration	0%	2 hrs	Fri 4/30/99	Fri 4/30/99		Feng Xun - INT
216	<b>QE</b>	<b>0%</b>	<b>18 hrs</b>	<b>Fri 4/23/99</b>	<b>Tue 5/4/99</b>		

ID	Task Name	% Complete	Work	Start	Finish	redecessor	Resource Names
217	<b>Test Plan</b>	<b>0%</b>	<b>8 hrs</b>	<b>Fri 4/23/99</b>	<b>Wed 4/28/99</b>	<b>206</b>	
218	QE Test Draft	0%	4 hrs	Fri 4/23/99	Fri 4/23/99		Yang An (Andre) - QE
219	QE Test Review & Update	0%	4 hrs	Fri 4/23/99	Wed 4/28/99	218	Yang An (Andre) - QE
220	<b>Tests</b>	<b>0%</b>	<b>10 hrs</b>	<b>Fri 4/30/99</b>	<b>Tue 5/4/99</b>	<b>214,217</b>	
221	New Feature Testing	0%	6 hrs	Fri 4/30/99	Fri 4/30/99		Yang An (Andre) - QE
222	Incorporation into Regression	0%	4 hrs	Tue 5/4/99	Tue 5/4/99	221	Yang An (Andre) - QE
223	<b>Total Dimension VA</b>	<b>0%</b>	<b>25 hrs</b>	<b>Fri 4/30/99</b>	<b>Thu 5/6/99</b>	<b>214</b>	
224	<b>Design</b>	<b>0%</b>	<b>2 hrs</b>	<b>Fri 4/30/99</b>	<b>Fri 4/30/99</b>		
225	Design Review	0%	1 hr	Fri 4/30/99	Fri 4/30/99		Feng Xun - DES
226	Engineering Review	0%	1 hr	Fri 4/30/99	Fri 4/30/99	225	Feng Xun - DES
227	<b>Implementation</b>	<b>0%</b>	<b>4 hrs</b>	<b>Fri 4/30/99</b>	<b>Fri 4/30/99</b>	<b>224</b>	
228	Reimplement code	0%	4 hrs	Fri 4/30/99	Fri 4/30/99		Feng Xun - IMP
229	<b>Standard work</b>	<b>0%</b>	<b>1 hr</b>	<b>Tue 5/4/99</b>	<b>Tue 5/4/99</b>	<b>227</b>	
230	Memory Leak, Diagnostics work	0%	1 hr	Tue 5/4/99	Tue 5/4/99		Feng Xun - SW
231	<b>Integration</b>	<b>0%</b>	<b>0 hrs</b>	<b>Tue 5/4/99</b>	<b>Tue 5/4/99</b>	<b>229</b>	
232	End to End Integration	0%	0 hrs	Tue 5/4/99	Tue 5/4/99		Feng Xun - INT
233	<b>QE</b>	<b>0%</b>	<b>18 hrs</b>	<b>Tue 5/4/99</b>	<b>Thu 5/6/99</b>		
234	<b>Tests</b>	<b>0%</b>	<b>18 hrs</b>	<b>Tue 5/4/99</b>	<b>Thu 5/6/99</b>	<b>231</b>	
235	New Feature Testing	0%	16 hrs	Tue 5/4/99	Thu 5/6/99		Busse Stephane - QE
236	Incorporation into Regression	0%	2 hrs	Thu 5/6/99	Thu 5/6/99	235	Busse Stephane - QE
237	<b>SQL Function Type</b>	<b>11%</b>	<b>76 hrs</b>	<b>Tue 3/30/99</b>	<b>Tue 4/20/99</b>		
238	<b>Design</b>	<b>100%</b>	<b>8 hrs</b>	<b>Tue 3/30/99</b>	<b>Wed 3/31/99</b>		
239	Design Review	100%	6 hrs	Tue 3/30/99	Wed 3/31/99		Zhang Parker - DES
240	Engineering Review	100%	2 hrs	Wed 3/31/99	Wed 3/31/99	239	Zhang Parker - DES
241	<b>Implementation</b>	<b>0%</b>	<b>32 hrs</b>	<b>Wed 3/31/99</b>	<b>Wed 4/7/99</b>	<b>238</b>	
242	Conversion	0%	16 hrs	Wed 3/31/99	Fri 4/2/99		Zhang Parker - IMP
243	Populating usage	0%	16 hrs	Fri 4/2/99	Wed 4/7/99	242	Zhang Parker - IMP
244	<b>Standard work</b>	<b>0%</b>	<b>16 hrs</b>	<b>Wed 4/7/99</b>	<b>Fri 4/9/99</b>	<b>241</b>	
245	Memory Leak, Diagnostics work	0%	8 hrs	Wed 4/7/99	Thu 4/8/99		Zhang Parker - SW
246	Code Review	0%	8 hrs	Thu 4/8/99	Fri 4/9/99	245	Zhang Parker - SW
247	<b>Integration</b>	<b>0%</b>	<b>8 hrs</b>	<b>Tue 4/13/99</b>	<b>Fri 4/16/99</b>	<b>244</b>	
248	Backend Integration	0%	4 hrs	Tue 4/13/99	Wed 4/14/99		Zhang Parker - INT
249	End to End Integration	0%	4 hrs	Thu 4/15/99	Fri 4/16/99	248	Zhang Parker - INT
250	<b>QE</b>	<b>0%</b>	<b>12 hrs</b>	<b>Fri 4/16/99</b>	<b>Tue 4/20/99</b>		
251	<b>Tests</b>	<b>0%</b>	<b>12 hrs</b>	<b>Fri 4/16/99</b>	<b>Tue 4/20/99</b>	<b>247</b>	
252	New Feature Testing	0%	8 hrs	Fri 4/16/99	Tue 4/20/99		Busse Stephane - QE

ID	Task Name	% Complete	Work	Start	Finish	redecessoi	Resource Names
253	Incorporation into Regression	0%	4 hrs	Tue 4/20/99	Tue 4/20/99	252	Busse Stephane - QE

9	Feb 14, '99							Feb 21, '99							Feb 28, '99							Mar 7, '99							Mar 14, '99							Mar 21, '99																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S

LI Benjamin Z. - DES

LI Benjamin Z. - DES, Yuan Jun - DES

Lao Rixun - DES

Wang Xu

Yuan

Ma

3/2

Bus





9	Feb 14, '99							Feb 21, '99							Feb 28, '99							Mar 7, '99							Mar 14, '99							Mar 21, '99								
T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T



9	Feb 14, '99							Feb 21, '99							Feb 28, '99							Mar 7, '99							Mar 14, '99							Mar 21, '99															
T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T

9	Feb 14, '99							Feb 21, '99							Feb 28, '99							Mar 7, '99							Mar 14, '99							Mar 21, '99						
T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T					

9	Feb 14, '99							Feb 21, '99							Feb 28, '99							Mar 7, '99							Mar 14, '99							Mar 21, '99															
T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T

9	Feb 14, '99							Feb 21, '99							Feb 28, '99							Mar 7, '99							Mar 14, '99							Mar 21, '99						
T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T					

Mar 28, '99							Apr 4, '99							Apr 11, '99							Apr 18, '99							Apr 25, '99							May 2, '99																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T





Mar 28, '99							Apr 4, '99							Apr 11, '99							Apr 18, '99							Apr 25, '99							May 2, '99																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										
F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T



Wang Xinyi - DES, Yuan Jun - DES, Feng Xun - DES, Li Benjamin Z. - DES

Yuan Jun - DES, Li Benjamin Z. - DES, Wang Xinyi - DES

Feng Xun - IMP

Feng Xun - IMP, Wang Xinyi - IMP

Feng Xun - SW, Wang Xinyi - SW, Ma Yuling - SW

Wang Xinyi - INT, Yuan Jun - INT, Feng Xun - INT, Ma Yuling - INT

Ma Yuling - IMP

Busse Stephane - QE

Busse Stephane - QE

Busse Stephane - QE

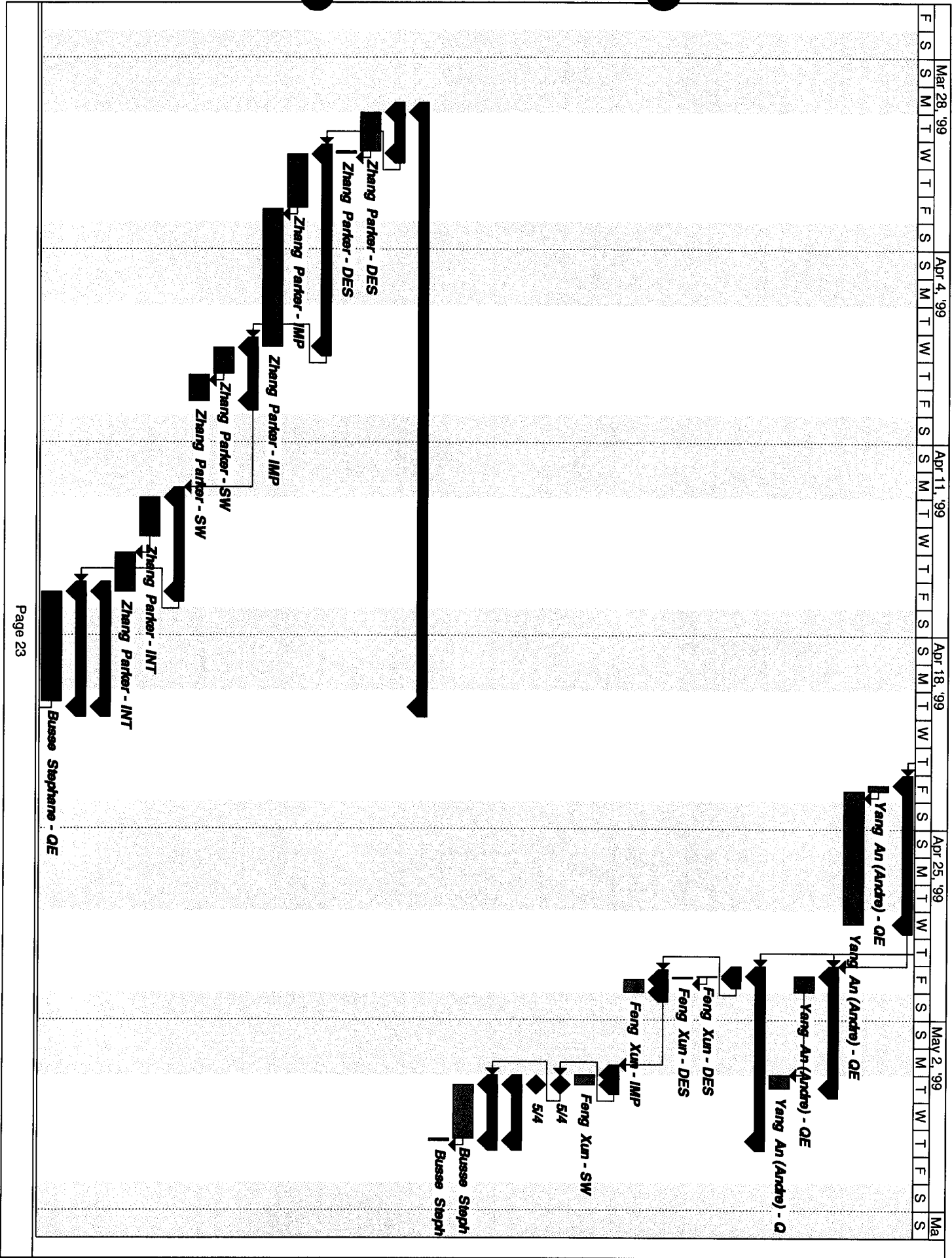
Busse Stephane - QE

Busse Stephane - QE

4/16

4/16











9, '99	May 16, '99	May 23, '99	May 30, '99	Jun 6, '99	Jun 13, '99	Jun 20, '99
M	S	S	S	S	S	S
T	M	M	M	M	M	M
W	T	T	T	T	T	T
T	W	W	W	W	W	W
F	F	F	F	F	F	F
S	S	S	S	S	S	S

9, '99	May 16, '99	May 23, '99	May 30, '99	Jun 6, '99	Jun 13, '99	Jun 20, '99
M	S	S	S	S	S	S
T	M	M	M	M	M	M
W	T	T	T	T	T	T
T	W	W	W	W	W	W
F	F	F	F	F	F	F
S	S	S	S	S	S	S



9, '99	May 16, '99	May 23, '99	May 30, '99	Jun 6, '99	Jun 13, '99	Jun 20, '99
M	S	S	S	S	S	S
T	M	M	M	M	M	M
W	T	T	T	T	T	T
T	W	W	W	W	W	W
F	T	T	T	T	T	T
S	F	F	F	F	F	F
S	S	S	S	S	S	S





[illegible]



## **Kemal Team Milestones – 1/30/98**

<b>Month</b>	<b>Objective</b>	<b>Status/Priority</b>
1/98	Integration: Round-trip report execution	Done
	Execution as a service	Done – Fragile
	Encryption of passwords	Done
	Installation routine	In progress
	DSS Net over HTTP	Done
	Performance monitoring: Local	Done
	Internationalization infrastructure	Done
2/98	Update resource server	Minimum
	Diagnostics: Enhance error handling and tracing	Minimum
	Ability to backup Server structures	Expected
	Ability to restore Server structures	Upside
	Finalize monitoring (except scheduled jobs)	Expected
	Login from Win 95 clients	Upside
	Project idling	Minimum
	Remote server administration	Minimum
	Job cancellation	Minimum
	Finalize Server internationalization	Upside
3/98	Configuration: Finalize run-time parameters	Minimum
	Performance monitoring: Remote	Minimum
	Job prioritization	Minimum
	Clustering	Expected
	Job statistics logging	Minimum
4/98	Load balancing	Expected
	Integrate datamarting capabilities	Expected
	Scheduling	Expected
5/98	Server optimizations	Expected
	Broadcaster integration	Expected

# **Castor Kernel Status**

*February 26, 1998*

## **Summary**

During the month of February, the Kernel team has focused on stabilization of the existing software and strengthening integration with other parts of the Server team. In addition, we invested time in re-examining our development process and identifying ways to improve our development as a whole. Specifically, the Kernel team has focused on closing outstanding issues found through testing, improving memory usage, and resolving issues that result from integration with COM and Engine modules. In addition to these tasks, the Server has added functionality including basic diagnostic functionality, and internationalization of Server modules.

From a quality engineering perspective, the Server team has focused on a scalability initiative. This effort focuses on providing the team the ability to monitor key scalability indicators as development proceeds. Specifically, the QE team has defined procedures for stressing the server and for monitoring memory leaks. These procedures include a set of quantifiable metrics that can be reported on a regular basis, as well as the testing infrastructure to do so (i.e. utility programs, etc.). Regular reporting of these metrics will serve as a key barometer of our development effort.

Finally, in documentation, we have begun work on conceptual material that will be included product manuals, as well as used to drive internal education on the Castor product suite.

## **Resources**

Luis Orozco, engineering management  
Scott Cappiello, program management

Wayne Li, engineering lead, Integration  
Kevin Wei, engineering, Integration

Sunil Dixit, engineering, Server Modules

Ramprasand Polana, engineering lead, Execution Pipeline  
Kaushal Sanghavi, engineering, Execution Pipeline

Ashish Soni, quality engineering lead  
Jianhua Wang, quality engineering  
Abdel Ghalayini, quality engineering

Randy Hechinger, documentation

## ***Feature Sets***

The following features sets are used to drive functional specifications, engineering designs, implementation planning, and test suites for the Castor Kernel. Unifying all of these functions under a common feature set is an important step for our development process because it ensures that each function on the team is using a common roadmap for their activities.

<b>ID</b>	<b>FEATURE SET</b>	<b>DESC</b>
1	Administration	Ability to administer server through the Server Admin API
2	Architecture	Features of the overall architecture that are not a specific feature set
4	Client Information	Capabilities of clients to request information from server
5	Configuration	Ability to set up and configure the server through the Admin API
6	Configuration-DBC's	Configuration of database connections
7	Configuration-Project	Configuration of projects within a server
8	Configuration-PU's & Tasks	Configuration of processing units within a server
9	Configuration-Queues & Priority	Configuration of queues within a server
10	Database Classes	Database-related features including connection caching
11	Diagnostics	All capabilities related to reporting errors and faults
12	Documentation	All documentation concerning server concepts and usage.
13	Fault Tolerance	Ability for Server to recover from errors and faults
14	Governing	Job governing parameters at all levels
15	Job Management	Server's internal management of jobs
16	Job Prioritization	Queues, queue sets, priority functions
17	Load Balancing	Performance optimizations within server
18	Monitoring	Ability to monitor jobs, users, and systems through the Admin API
19	Network	Communications with server
20	Object Server Integration	Server's interaction with the Object Server
21	Performance Monitor	Integration with the NT Performance Monitor
22	Processing Units	Operation of processing units and their component threads
23	Projects	Project definition relevant to server
24	Report Server Integration	Server's interaction with the Report Server
25	Resource Server Integration	Server's interaction with the Resource Server
26	Scheduling	Ability to schedule jobs for execution
27	Security	All aspects of security
28	Job Statistics	Logging of server-related execution statistics
29	Synchronization Classes	Classes that govern timing within the Server
30	User Management	Creation, grouping, and management of users.
99	Other	Miscellaneous features
31	Installation	All aspects related to the installation of server and its components
32	Datamarting	Features related to Server's ability to create and manage datamarts
33	Internationalization	Ability for the Server to support multiple locales
34	Web Integration	Features related to accessing Server via the web
35	COM Integration	Integration with modules from the COM team
36	Engine Integration	Integration with modules from the Engine team

## ***Milestones***

See spreadsheet, "Server Engineering Q1 Plan".



## ***Castor Kernel Status***

*March 27, 1998*

### ***Summary***

During the month of March, the Castor Kernel team saw modest progress on our implementation goals. The team completed some key areas of functionality, but overall progression through our plan is somewhat slower than expected.

We made a lot of progress in the implementation of our diagnostic strategy, ensuring the proper use of error codes within the server and enabling the redirection of error messages to a variety of output devices. While incorporating this diagnostics infrastructure, we also ensured that server modules are able to report messages in a locale-independent way, in accordance with our internationalization strategy. We also added the ability to load, unload, idle, resume, and monitor projects through our Server Admin API. In addition, we completed a research project investigating the use of DHTML to access the Castor Server. A prototype and research document were presented to the entire Castor team. Finally, we also began work on logging job statistics to a database and the ability to set the server's operational mode based on a schedule.

The Kernel quality engineering team continues to perform testing for Kernel-specific modules as well as testing for the integrated Castor Server. During the month, our quality engineering team has further formalized our process for monitoring memory consumption and stress capability in the Castor Server. A battery of tests is conducted against each weekly build with the results stored in TQMS. A DSS Broadcaster application is able to send reports out on a weekly basis.

A considerable amount of time during the month was spent on design and documentation activities. The senior engineers on the team are spending little time actually implementing features, but rather designing, documenting, and communicating with other teams. While this is probably an appropriate use of their time, it is having an effect on the amount of feature implementation the team can take on. In particular, significant design energy was focused on the Kernel's integration with the Resource Server, logging of job statistics, and the optimal management of database connections. The last topic (database connection management) was prompted in part by some high-end requirements expressed by Kmart during the month.

Members of the team also invested time in delivering technical training to other members of the Castor team. Engineers from the Kernel team led training sessions in the Castor diagnostics strategy and in memory leak detection techniques.

Looking forward to the April build, we expect to implement the third phase of our diagnostics strategy, which focuses on tracing capabilities. Also, we will work on logging job execution statistics to a database, allowing server to idle a project or set any governing parameter based on a calendar schedule, and ensuring that the necessary kernel components can run under Windows 95.

## ***Resources***

The resource roster for the Kernel team is shown below. Development is divided into three subteams. "Integration" is the interaction of the Kernel with modules from other teams, including the Server Admin API. "Execution pipeline" is the main execution of the Castor Server, which governs the main processing flow. "Server modules" are core infrastructure classes and modules that lie outside the execution pipeline.

Name	Role	Subteam	Allocation
Luis Orozco	Engineering manager		PT
Scott Cappiello	Program manager		PT
Wayne Li	Engineering lead	Integration	FT
Kevin Wei	Engineering	Integration	FT
Sunil Dixit	Engineering	Server modules	FT
Ramprasand Polana	Engineering lead	Execution pipeline	FT
Kaushal Sanghavi	Engineering	Execution pipeline	PT
Ashish Soni	Quality engineering lead		PT
Abdel Ghalayini	Quality engineering		FT
Jianhua Wang	Quality engineering		FT
Randy Hechinger	Documentation		PT

**Note:** The quality engineers on the Kernel team provide quality engineering for the Kernel team's modules as well as integrated Server QE (Kernel, COM, Engine).

## ***Issues***

- *Dependencies with other teams.* Implementation is slowed most often due to the subtle dependencies that exist between all the Castor teams. Naturally, we should remove unnecessary dependencies as much as possible. However, once each team has a critical mass of functionality, it will be worthwhile to have a more integrated feature-driven approach to implementation. We need to select features that cross all teams and drive them to completion across teams. Note that this will only work if it does not distract the individual teams from implementing core functionality.
- *Insufficient review of functional specifications and designs.* While reviewing implementation from a feature perspective, we are discovering some holes in the implementation, functional specification, and design of many features. We can continue to barrel through with implementation, but we risk having to revisit our work without sufficient review of specifications, designs, and implementation.
- *Planning risks.* Our plans currently run through the end of June and are quite aggressive. In addition, the plans shown here reflect implementation alone. The plans do not account for time spent on stabilization and support development (e.g. internal training, migration utilities, beta programs) are not included.
- *Castor Web direction.* To date, the Kernel team has invested time in researching technologies relevant to our Web strategy. We need to determine the Web direction for Castor so we can determine if our current plans will be affected.
- *Fault tolerance lab.* The Kernel team still needs to acquire hardware in April for fault tolerance work scheduled in May.

### ***Feature Summary: March***

The features listed below were implemented during March 1998.

<b>Feature Set</b>	<b>Feature</b>	<b>Description</b>
Client Information	Registered Servers and Projects	Ability for clients to see all registered servers and the list of projects available via those servers.
Diagnostics	Router Class – Server	Incorporate router class in Server internally
Diagnostics	Router Class – Admin API	Incorporate router class in Server Admin API
Diagnostics	VB Compliant router class	Router class can be used by VB components.
Diagnostics	Error Handling: Server	Error handling standard adopted by Server internally
Diagnostics	Infrastructure Classes	Complete infrastructure classes (log, router, NT event, debug) for diagnostics.
Internationalization	Server internationalization	Internationalization of messages within Server
Internationalization	Admin API internationalization	Internationalization of messages within the Server Admin API
Job Management	Thread servicing	Apply thread service policies
Monitoring	Project status	Ability to monitor the status of a project
Monitoring	Jobs - browsing info	Ability to monitor Jobs (condensed info)
Resource Server Integration	Resource Server Design	Design of Resource Server to be passed to COM team for implementation.
Web Integration	DHTML prototype	Prototype tree control, grid control, and drag and drop via DHTML.

### ***First Quarter Summary***

The features listed below were implemented during or prior to Q1 1998.

<b>Feature Set</b>	<b>Feature</b>	<b>Description</b>
Administration	Local startup/shutdown	Ability to startup/shutdown Server locally as an application
Administration	Project idle modes: Full idle	Ability to idle and resume server by project (full idle)
Administration	Idle after current jobs finish	Ability to idle a project after waiting for all current jobs to execute
Administration	Project idle modes: Client idle	Ability to idle project from client requests
Administration	Remove job	Ability to remove a job through the Admin API
Administration	Disconnect user	Ability to disconnect a user from Server through the Admin API
Administration	Immediate shutdown sequence	Ability to shutdown the server immediately
Administration	Project loading/unloading	Ability to take a project online/offline
Administration	Project idle modes: DB idle	Ability to idle project from job execution
Architecture	Server as Service	Ability to run the server as an NT service.
Client Information	Jobs in Queue	Ability for clients to view all their reports currently in DSS Server job queue.
Client Information	Individual Job Retrieval	Ability for clients to retrieve individual jobs.
Client Information	Individual Job Statistics	Ability for clients to view execution statistics for an individual job.
Configuration	Service configuration	Ability to configure server to run as a service
Database Classes	Execution mode	Ability to execute queries asynchronously or synchronously
Database Classes	Multithreaded mode	Ability to run in multithreaded mode
Database Classes	ODBC connectivity	Ability to use ODBC connectivity
Database Classes	Double Byte Databases	Ability to work with Double Byte databases
Database Classes	Multiprocess mode	Ability to run in multiprocess mode
Diagnostics	Locking violations	Ability to track locks by thread
Job Management	Unique Job ID's	Ability to assign unique job ids
Job Management	View SQL	Ability to view a job's SQL
Job Management	View errors	Ability to view any job execution errors on client side or from Administrator tool
Job Management	Job close	Ability to close a job
Job Management	Job status	Ability to retrieve the status of an individual job or group of jobs
Monitoring	Jobs - detail info	Ability to monitor Jobs (complete info)
Monitoring	User connections	Ability to monitor user connections within server
Monitoring	Server status	Ability to monitor the current status of a server (started, stopped)
Monitoring	Jobs – SQL	Ability to monitor the SQL associated with a job
Monitoring	Jobs - error message	Ability to monitor error messages associated with a job
Network	TCP/IP	Ability to support TCP/IP
Network	Admin requests	Ability to accept administrative requests
Network	Job requests	Ability to accept job requests

Network	Asynchronous service	Ability to serve requests asynchronously
Network	HTTP	Ability to support HTTP
Object Server Integration	Basic integration	Server uses Object Server to manage schema and application objects
Performance Monitor	Local NT integration	Ability to monitor server statistics through the NT Performance Monitor.
Performance Monitor	Network Counters	Performance Monitor tracks information about network requests
Performance Monitor	Job Counters	Performance Monitor tracks job information
Performance Monitor	Database Counters	Performance Monitor tracks database connection information
Performance Monitor	User Counters	Performance Monitor tracks user information
Performance Monitor	Miscellaneous Counters	All other counters described in the functional specification
Projects	Multiple Projects	Ability to run multiple projects on one DSS Server machine.
Report Server Integration	Basic integration	Server uses Report Server to execute reports
Security	Encryption	Ability to encrypt all sensitive data in files and network
Security	NT integration	Ability to use NT security system for password validation
Security	Database Privileges	Ability to map users to database logins
Security	Pass Through	Ability to use database for password validation
Security	Client Encryption	All passwords encrypted when sent from the client
Synchronization Classes	Robust memory	Ability to effectively run in multithreaded mode without experiencing memory corruptions
Synchronization Classes	Deadlock prevention	Ability to effectively run in multithreaded mode without running into deadlocks
Synchronization Classes	Exception handling	Ability to roll back locks in case of exceptions
User Management	Login Validation	Ability to verify user account within two tier or three tier environment.

***Current Plan: April Build***

<b>Build Month</b>	<b>Feature Set</b>	<b>Feature</b>
APR	Administration	Operational schedule
APR	Administration	Statistics integration
APR	Configuration	Diagnostic configuration
APR	Configuration	Thread servicing scheme
APR	Diagnostics	Error Handling: Admin API
APR	Diagnostics	COM Integration
APR	Diagnostics	Engine Integration
APR	Diagnostics	Boundary tracing
APR	Diagnostics	Startup diagnostics
APR	Diagnostics	Thread tracing
APR	Diagnostics	Network tracing
APR	Diagnostics	Job Cancel tracing
APR	Diagnostics	Profiling
APR	Diagnostics	User connection tracing
APR	Diagnostics	Job Execution tracing
APR	Diagnostics	Job ID tracing
APR	Diagnostics	Debug monitor
APR	Diagnostics	NT Event Log integration
APR	Diagnostics	Error logging
APR	Diagnostics	Configuration tracing
APR	Diagnostics	Database tracing
APR	Diagnostics	Basic Internationalization
APR	Monitoring	Summary information
APR	Projects	Project registration
APR	Resource Server Integration	Server integration
APR	Security	Win95 Login - Three-tier
APR	Security	Two-tier encryption
APR	Job Statistics	Server statistics
APR	Job Statistics	Statistics configuration
APR	Job Statistics	SQL configuration
APR	Job Statistics	Job submission
APR	Job Statistics	User connections
APR	Job Statistics	Job Source
APR	Job Statistics	View statistics
APR	Job Statistics	WH Monitor integration
APR	Web Integration	Connection pooling

## ***Current Plan: May Build***

<b>Build Month</b>	<b>Feature Set</b>	<b>Feature</b>
MAY	Administration	Alter priority
MAY	Administration	Governing integration
MAY	Administration	Run time parameters
MAY	VLDB Optimizations	Drop table modes
MAY	VLDB Optimizations	Database per thread
MAY	VLDB Optimizations	Database login per thread
MAY	Configuration	Server Governors
MAY	Configuration	Server-User Governors
MAY	Configuration	Server-Project Governors
MAY	Configuration	Server-Project-User Governors
MAY	Configuration	Catalog Locking Workarounds
MAY	Configuration	Change DSN
MAY	Configuration-Project	User access
MAY	Diagnostics	International log viewer
MAY	Documentation	Server Concepts
MAY	Fault Tolerance	Backup
MAY	Fault Tolerance	Restore
MAY	Fault Tolerance	Clustering
MAY	Fault Tolerance	Alert notification
MAY	Governing	Server level
MAY	Governing	Server/User level
MAY	Governing	Server/Project level
MAY	Governing	Server/Project/User level
MAY	Governing	Server governors
MAY	Governing	Project governors
MAY	Governing	User governors
MAY	Governing	Shared login governors
MAY	Governing	User connection governors
MAY	Governing	Job governors
MAY	Governing	Order of precedence
MAY	Governing	Default governors
MAY	Job Management	Incremental Fetching
MAY	Job Management	Job retrieval
MAY	Job Management	Job push back
MAY	Job Management	Job timeout
MAY	Job Management	Job kill
MAY	Job Management	Job cleanup
MAY	Job Management	Close old jobs
MAY	Job Management	Alter priority
MAY	Job Prioritization	User priority
MAY	Job Prioritization	Cost priority
MAY	Job Prioritization	Project priority
MAY	Job Prioritization	Alter priority
MAY	Job Prioritization	Priority formula
MAY	Load Balancing	Intra-unit throughput

MAY	Load Balancing	Inter-unit throughput
MAY	Monitoring	Database connections
MAY	Job Statistics	Job processing
MAY	Job Statistics	Job events
MAY	Job Statistics	User sessions
MAY	Job Statistics	User events
MAY	Job Statistics	Selective Purge
MAY	Datamarting	All Datamarting
MAY	Testing Automation	Database emulator
MAY	Function Server Integration	Integration

***Current Plan: June Build***

<b>Build Month</b>	<b>Feature Set</b>	<b>Feature</b>
JUNE	Administration	Logging integration
JUNE	Administration	Broadcast message
JUNE	Client Information	Cache Status
JUNE	Diagnostics	Install diagnostics
JUNE	Diagnostics	Scheduling tracing
JUNE	Diagnostics	Unique log file names
JUNE	Documentation	Server Configuration
JUNE	Documentation	What's new
JUNE	Governing	Governing schedule
JUNE	Monitoring	Scheduled tasks
JUNE	Performance Monitor	Remote monitoring
JUNE	Performance Monitor	Admin API integration
JUNE	Performance Monitor	Internationalization
JUNE	Scheduling	Basic scheduling
JUNE	Scheduling	Event-based scheduling
JUNE	Scheduling	Monitoring
JUNE	Scheduling	Change properties
JUNE	Job Statistics	Table Hits
JUNE	User Management	Timer class



## ***Castor Kernel Status***

*May 31, 1998*

### ***Summary***

#### ***Quality achievements***

- *Established Systems Integration team.* Following University Week, resources from the Kernel team were reallocated to form a new task force on the Castor Server effort: the Systems Integration team. This team is responsible for implementation and integration issues within the integrated Castor Server that do not have clear ownership.
  - *Integrated quality efforts across the castor Server teams.* Luis Orozco is focusing on managing the integrated Castor QE effort.
- *Improved build process.* After Company Days, Significant effort to improve effectiveness of build process.
- *Significantly reduced memory leaks.* While a portion of the development teams focused on design work during the month of May, other team members focused on stabilizing the existing state of the software. Significant progress was made on memory leaks, through formalizing the process of reporting leaks and focused efforts to resolve memory leak issues.

#### ***Design achievements***

- *Focused on designs.* All Castor Server teams targeted outstanding designs throughout the month, especially those crossing development teams. This allows us to identify cross-team dependencies early in our development process and to plan accordingly. The Kernel and Systems Integration teams made significant progress in the following designs:
  - Job prioritization and servicing
  - Scheduling
  - Backup and Restore
  - Progress notification
  - Client login personalization
  - Database management

#### ***Implementation achievements***

- *Developed set of integrated Server goals.* This feature set will allow the Castor Server team to monitor progress and achieve small development victories around key cross-team features. The plan is in a living document located at [\\tech1\castor\plan\server](#).
- *Implemented Server components on Windows 95.* This verifies our ability to run the Server components required by a Windows 95 client.
- *Implemented asynchronous report execution.*

## ***Resources***

The resource roster for the Kernel team and the Systems Integration team are shown below. Note that the quality engineers on the Kernel team provide quality engineering for the Kernel team's modules as well as integrated Server QE (Kernel, COM, Engine, Systems Integration).

Name	Role	Subteam	Allocation
Luis Orozco	Engineering manager		PT
Scott Cappiello	Program manager		PT
Wayne Li	Engineering lead	Systems Integration	FT
Kevin Wei	Engineering	Systems Integration	FT
Ningning Liu	Engineering	Systems Integration	FT
Janaki Goteti	Engineering	Systems Integration	FT
Ramprasand Polana	Engineering lead	Kernel	FT
Sunil Dixit	Engineering	Kernel	FT
Kaushal Sanghavi	Engineering	Kernel	PT
Ashish Soni	Quality engineering lead		PT
Abdel Ghalayini	Quality engineering		FT
Jianhua Wang	Quality engineering		FT
Randy Hechinger	Documentation		PT

## ***Issues***

- *Dependencies with other teams.* Implementation is slowed most often due to the subtle dependencies that exist between all the Castor teams. Naturally, we should remove unnecessary dependencies as much as possible. We are doing three things to address this. First, we have established the Systems Integration team to drive integration issues to completion that might otherwise not have a clear owner. Second, we front-loaded design activities, especially those that cross teams, to establish a clearer understanding of dependencies that may arise. Finally, we have assembled a set of integrated features that we will track as small victories on the road to Company Days and to Phase I Alpha.
- *Ability to resolve stress- and performance-related issues.* The Systems Integration team and quality engineers have been working on infrastructure to stress the Castor Server and apply tests to gauge performance-related information. To date, it has been difficult to resolve bugs that are logged in these situations. Part of the effort moving forward will be to make it easier to track down and resolve issues in this category.
- *Non-product tasks.* We are beginning to get a better handle on our planning for feature design and implementation, although our plans are still aggressively scheduled through August. However, the plans do not account for time spent on stabilization and support development (e.g. internal training, migration utilities, beta programs) are not included. We are addressing the issue with stabilization with Luis Orozco's effort to provide integrated planning and management of the quality process across teams. We are addressing the issue with support development by working more closely with External Education and identifying a Castor Beta program manager, but this is in the early stages.
- *Fault tolerance lab.* The hardware for our fault tolerance design and implementation is somewhere within MicroStrategy, but the Kernel team is not yet able to use it.

## **Castor Kernel Status**

August 10, 1998

### ***Brief Summary***

#### ***Progress towards Phase 0 goals***

- *ZDB.* We are still a significant way off of our quality goals. Many issues are system-level issues that are slow to troubleshoot or issues related to functionality that is not complete.
- *Server and project configuration.* We need to update the server installation and run through the end-to-end configuration scenario to confirm that this works. The goal is to be able to do a clean install of server and get a project available for use.
- *Installation.* We want the installation of Office to be available via the Web. The Server installation will be covered under the configuration scenario mentioned above. Web configuration will be manually-intensive, as we have not yet implemented Web administration features.

### ***Issues***

- *Resources who have Abell duties.* The tasks are impacting progress on Castor.
- *System-level issue resolution.* These issues take more time than expected to resolve.

### ***Next Steps***

#### ***Alpha 1***

- Primary features to implement.
  - Scheduling
  - Statistics
  - Backup and restore
  - Unify job monitoring
- Focus on system-level issues. We will likely allocate developers to focus on system-level issues rather than new feature development.

### ***Resources***

The resource roster for the Kernel team and the Systems Integration team are shown below. Note that the quality engineers on the Kernel team provide quality engineering for the Kernel team's modules as well as integrated Server QE (Kernel, COM, Engine, Systems Integration).

Name	Role	Subteam	Allocation
Luis Orozco	Engineering manager		PT
Scott Cappiello	Program manager		PT
Wayne Li	Engineering lead	Systems Integration	FT
Kevin Wei	Engineering	Systems Integration	FT
Ningning Liu	Engineering	Systems Integration	FT
Janaki Goteti	Engineering	Systems Integration	FT
Ramprasand Polana	Engineering lead	Kernel	FT
Sunil Dixit	Engineering	Kernel	FT
Kaushal Sanghavi	Engineering	Kernel	PT (until 8/30)
Ashish Soni	Quality engineering lead		PT
Abdel Ghalayini	Quality engineering		FT
Jianhua Wang	Quality engineering		FT
Randy Hechinger	Documentation		PT

# **Castor Kernel Status**

August 28, 1998

## ***Summary***

### ***Achievements during Phase 0***

- *ZDB.* The primary achievement over the past month was realization of the Phase 0 ZDB. The team actually closed down all issues eligible for the ZDB.
- *Memory leaks.* The effort to drive memory leaks to an acceptable Phase 0 range (<3 MB per 1000 report jobs) was driven by the Systems Integration team.
- *Major features for Phase 0.* The features below represent significant functionality of the Phase 0 Castor kernel.
  - Server installation and configuration utility.
  - Installation of Office and Server available over the Web.
  - Server and project configuration.
  - Monitoring of jobs and users.
  - Job cancel and cleanup.
  - Administration of project idle modes.
  - Web support: communication over HTTP.
  - Error codes bubble up from low-level components.
  - Ability to route error messages and traces to a variety of output devices.
  - Server-level and some project-level governing parameters.
- *Server and project configuration.* A major goal of the Phase 0 milestone was to achieve a consistent end-to-end story, including the ability to install, configure, and start the server from scratch, then define a project and run a report. This represents a significant integration achievement.

## ***Issues***

- *Resources who have Abell duties.* The effectiveness of the team overall is hampered by tasks that require individuals to divert their attention from Castor.
- *Server Admin GUI coordination.* The Kernel team management is principally responsible for delivering backend functionality. More time is required to spend with the Server Admin GUI team to coordinate effort on end-to-end feature development.
- *Need for stability and performance metrics.* The implementation has now reached a stage where we can begin to accurately measure metrics related to performance and availability. As we do so, we will uncover issues that need developers' attention. More time from Kernel developers will be shifted to resolving such issues than implementing new features.

## ***Next Steps***

### ***Implementation Plans***

Please see the integrated Kernel, COM, Engine, and SI development plans for the feature outlook over the next two months.

## ***Resources***

The resource roster for the Kernel team and the Systems Integration team are shown below. Note that the quality engineers on the Kernel team provide quality engineering for the Kernel team's modules as well as integrated Server QE (Kernel, COM, Engine, Systems Integration).

Name	Role	Subteam	Allocation
Luis Orozco	Engineering manager		PT
Scott Cappiello	Program manager		PT
Wayne Li	Engineering lead	Systems Integration	FT
Kevin Wei	Engineering	Systems Integration	FT
Ningning Liu	Engineering	Systems Integration	FT
Janaki Goteti	Engineering	Systems Integration	FT
Ramprasand Polana	Engineering lead	Kernel	FT
Sunil Dixit	Engineering	Kernel	FT
Ashish Soni	Quality engineering lead		PT
Abdel Ghalayini	Quality engineering		FT
Jianhua Wang	Quality engineering		FT
Randy Hechinger	Documentation		PT

## **Castor Kernel Status**

September 25, 1998

### ***Summary***

#### ***Phase 0 "Chinese" Milestone***

The Phase 0 Chinese milestone represents an internal engineering milestone for conclusion of key pieces of functionality, most notably support for multi-byte and Unicode environments. We have had serious difficulty in actually closing out this milestone.

- *No P1 issues.* This actually has proven to be the most challenging objective to reach. See issues, below.
- *Memory leaks.* The effort to drive memory leaks to an acceptable Phase 0 range ( $< 0.4$  MB per 1000 report jobs) is being driven by the Systems Integration team.
- *Major features for Phase 0 Chinese.* The features below represent new functionality during this milestone.
  - *Backup and restore.* If the server fails for any reason, it can restore the jobs that had been submitted prior to the failure.
  - *Temp table cleanup at project startup.* When a server restarts a project, it cleans up any temp tables left over due to server failure.
  - *Database login mapping (backend only).* Different DSS Users can be mapped to access the warehouse using different database logins.
  - *Database connection monitoring for MD6 projects (including frontend).* The Server Administrator shows the connections that the DSS Server opens against the database to access queries.
- *Performance testing infrastructure.* The QE team developed a Performance 1000 test that can be used to monitor Server throughput, much like our Memory 1000 test that monitors memory leaks on a daily basis. This infrastructure allows us to attach throughput objectives to our milestones, just like we have established memory leak objectives to milestones in the past.

#### ***October 23 Milestone***

October 23 represents the external Alpha milestone for the entire Castor team. We have structured our objectives so that implementation for all features visible through the GUI is complete by September 30. After this date, we will turn the Kernel team towards stability, performance, and scalability. Specifically, we expect that our testing at this stage will reveal a number of issues that require considerable attention to close and resolve. Our resources will be the first to focus on closing down issues for October 23, as we expect our issues to take the longest to resolve.

To date we are on track to wrap up by 9/30 the development of the backend portion of the following features:

- *Inbox.* This will allow users to retrieve report results for reports that completed while they were logged out of Office.
- *Statistics.* The server will log statistics about report jobs and user connections to a relational database. We expect this to be helpful as we tune the server for performance.
- *Project configuration.* There is a set of functionality with the Server Administrator that will allow us to map DSS users to different database logins, define logical database definitions completely, set project-level governing parameters, and configure the logical databases used by a project.
- *Web server administration.* This will allow us to open, close, and configure the 4-tier gateway to the server.
- *Failover with server recovery.* This will allow us to support recovery of the server when running in a clustered configuration using Microsoft Cluster Server. When the failed server is restarted, it can restore the state of the previously running server.

In addition, the Systems Integration team is completing work on Report-level Caching.

### ***Issues***

- Difficulty in closing out the Phase 0 Chinese milestone. I believe there are two main reasons for the difficulty we experienced. First, we did not establish a build process to complete this milestone early enough. We were trying to close out issues for a milestone on the same branch of code where new features were being introduced. In addition, we were slowed by issues that appeared because we have not yet set up the infrastructure to allow local builds that can catch issues before they are introduced at the team level. Second, the attention of the team was diluted by the urgency of the October 23 deadline. While the team considers the Chinese milestone to be important, we could not escape the planning and preparation required for meeting the October 23 milestone.

### ***Next Steps***

#### ***Implementation Plans***

- *Preparation for October 23.* As mentioned above, we are closing in on the feature implementation objectives for September 30. Afterwards, we will focus on integration and system level testing and robustness.
- *Performance measurement.* Currently, our testing has shown that the Server supports throughput of about 250 jobs per minute, with no caching. Although this is not a strictly apples-to-apples comparison to the Abell architecture, we interpret this to be slightly better than what Abell is capable of. Our target for October 23 is to reach 500 jobs per minute on a single processor machine. By Beta, we would expect to reach 800 jobs per minute. This is an interesting number for us because it represents 1 million jobs per day. Obviously, we need to refine these objectives to account for hardware constraints, usage profiles, and the use of caching; these estimates are meant to serve as general targets.

### ***Resources***

The resource roster for the Kernel team and the Systems Integration team are shown below. Note that the quality engineers on the Kernel team provide quality engineering for the Kernel team's modules as well as integrated Server QE (Kernel, COM, Engine, Systems Integration).

Name	Role	Subteam	Allocation
Luis Orozco	Engineering manager		PT
Scott Cappiello	Program manager		PT
Wayne Li	Engineering lead	Systems Integration	FT
Kevin Wei	Engineering	Systems Integration	FT
Ningning Liu	Engineering	Systems Integration	FT
Janaki Goteti	Engineering	Systems Integration	FT
Ramprasand Polana	Engineering lead	Kernel	FT
Sunil Dixit	Engineering	Kernel	FT
Ashish Soni	Quality engineering lead		PT
Abdel Ghalayini	Quality engineering		FT
Jianhua Wang	Quality engineering		FT
Randy Hechinger	Documentation		PT
David Weld	Quality engineering		FT

# ***Castor Server Status: Kernel and Server Administrator***

*November 23, 1998*

## ***Summary***

### ***Alpha 2 Progress***

Key new features for Castor Server in the Alpha 2 release:

- *Scheduling.* We are implementing the Castor version of scheduling, which includes support for “true scheduling”: the ability to specify that a report be submitted at a specific time. This phase of scheduling will focus on report scheduling. Our architecture has been designed to support the scheduling of administrative requests using the same scheduling mechanisms, but support for this will not be available in Alpha 2.
- *Job Prioritization and Servicing.* We are implementing the first stage of our new prioritization and servicing design. In the Alpha 2 release, users will be able to prioritize reports based on project, user, or a user-specified report cost.
- *Report Caching.* While the Alpha 1 release supports memory-based report caching, the Alpha 2 release will extend this to file-based caching.
- *Failover.* We will continue to enhance our support for failure recovery of the server when running in a clustered configuration using Microsoft Cluster Server. When the failed server is restarted, it can restore the state of the previously running server.

Key new features for Server Administrator in the Alpha 2 release:

- *Schedule monitoring.* Server administrators will be able to monitor the scheduled report requests that the server maintains.
- *Job Prioritization and Servicing.* This is the GUI component corresponding to the feature described above.
- *Cache monitoring.* This is an aggressive target for us, but we are planning to be able to monitor report caches through the Server Administrator, similar to the functionality of DSS Server 5.5's Cache Manager.
- *Cache administration.* In addition to monitoring caches, we will also support a few administrative capabilities, such as the ability to remove a cache.
- *Database login editor.* This functionality was left over from Alpha 1, but its inclusion rounds out the functionality of our “Database Definition Editors”. This set of editors includes all the functionality required to let system administrators define database connectivity, share them across servers and projects, map users to various database logins, etc.

## ***Next Steps***

### ***Feature Implementation***

- *Preparation for January 22.* As mentioned above, we are working towards implementation of a set of new features for Alpha 2. One of the biggest challenges for us will be ensuring adequate productivity through the holiday season. Our plans have been structured to accommodate the expected vacations of the coming months.



## ***Performance Measurement***

- ***Current DSS Server Performance.*** Currently, our testing has shown that the Server supports throughput of up to about 300 jobs per minute, with no caching. Although this is not a strictly apples-to-apples comparison to the Abell architecture, we interpret this to be slightly better than what Abell is capable of. Our target for January 22 is to reach 800 jobs per minute on our 4-processor performance testing machine. This number is a good course indicator because it represents the overhead involved with DSS Server processing, with the effects of caching and database response time removed. It is also an interesting number for us because it represents 1 million jobs per day.
- ***Testing and measurement infrastructure.*** Our performance testing efforts are currently focused on “architectural tuning”. We are trying to identify bottlenecks in the general execution cycle and determine which modules represent the best opportunity for tuning. Our current test suites focus on the following areas:
  - Differences between SQL generation, SQL execution, and crosstabbing.
  - Differences between synchronous and asynchronous execution.
  - Effects of caching.
  - Effects of using statistics logging.
  - Effects of using diagnostics logging.

## ***Resources***

The resource roster for the Castor Server team is shown below.

Name	Role	Subteam	Allocation
Wayne Li	Engineering manager		FT
Kevin Wei	Engineering	Systems Integration	FT
Ningning Liu	Engineering	Systems Integration	FT
Janaki Goteti	Engineering	Systems Integration	FT
Sunil Dixit	Engineering	Kernel	FT
Ramprasand Polana	Engineering lead	Kernel	FT
Ashish Soni	Quality engineering lead		PT
Jianhua Wang	Quality engineering lead		FT
Abdel Ghalayini	Quality engineering		FT
David Weld	Quality engineering		FT
Randy Hechinger	Documentation		PT
Scott Cappiello	Program management		PT

## ***Castor Server Status: Kernel and Server Administrator***

*December 21, 1998*

### ***Summary***

Over the past month, the entire Castor team has been focused on development of features for our Alpha 2 milestone. We have formed a number of feature teams that cross the familiar product-oriented teams in order to concentrate on delivery of the *feature*. The Kernel team had an active role in new features including report caching, scheduling, report subsetting, job prioritization and servicing, inbox, and report views. While completing this backend functionality, the Server also worked with the GUI teams to complete frontend work for prioritization and servicing, report schedule creation, schedule events creation, and diagnostics configuration.

In addition to work related to these features, our quality engineering efforts have been focused on controls to help us monitor memory leaks, performance, and platform compatibility of the Castor Server. Many people will be interested in our performance analysis to date. In these tests, we are currently focusing on minimizing the overhead introduced by our architecture. We use a weekly scorecard to monitor report job throughput at a high level. This scorecard pulls all external factors out of the equation (e.g. database execution time) by running reports that return very quickly. The maximum throughput that we are achieving is around 375 report jobs per minute; which translates into about half a million jobs per day on a single server. With the current implementation of caching, we can achieve around 500 report jobs per minute. The machine we use for testing is a 4-proc box with fairly slow processors (~200 MHz). As we finalize the features for Alpha 2, we are expanding the scenarios for performance testing to generate numbers that reflect real-world usage (i.e. mixture of report complexity, cache hit ratios, non-report activity, etc.).

Also, we have run some "endurance" tests on the same configuration in which we have pushed through over two million simple report executions through the server over a period of a few days. Again, these results are achieved under simplified scenarios that do not necessarily reflect real-world conditions, but the results are encouraging nonetheless. We are beginning to step up the degree of difficulty for the server in these tests, including running million-job endurance tests that include metric qualifications and ranking reports and running result sets with more than one million rows returned.

### ***Demonstration Topics***

Server Admin: Job Prioritization and Servicing

### ***Milestone Schedule***

Milestone	Description	Target Date
Alpha 2 ZDB	Quality milestone to coincide with Q1 Company Day. This ZDB will be taken to a set of customers for external testing.	1/22/1998
Alpha 3 ZDB	Feature complete for Server, API, and design tools. This ZDB will be available for customers for external testing.	4/2/1998

## ***Status***

### ***Achievements of the past month***

#### **Feature development towards Alpha 2**

- *Inbox.* Sergio Trejo and Ramprasad Polana completed implementation of core Inbox functionality. Using the Inbox, if a user submits a report that completes while the user is disconnected, a message appears in the Inbox to let the user know that the results are available. When combined with report caching, the user is able to retrieve results directly from the Inbox message.
- *Report caching.* Ningning Liu has been focused on using caching in the report execution cycle. Report caches allow result sets to be saved in memory and to disk so those queries do not have to be re-executed against the warehouse database. For Alpha 2, we have focused on adding file-based caches.
- *Report subsetting.* Ningning Liu has also been working on subsetting functionality, along with members of the Engine and COM teams. Subsetting allows reports to use partial results of existing caches. This will be an important feature for Broadcaster personalization. For Alpha 2, the Server supports template subsetting. Filter subsetting will be added later.
- *Report views.* The Castor Server also supports multiple simultaneous views of the same report. This will allow a report viewer to display the same result set in multiple ways at the same time (e.g. a grid and a graph, for instance).
- *Scheduling.* Andres Paz, Janaki Goteti, and Ramprasad Polana have put in place the first iteration of the Castor Server scheduling functionality. For Alpha 2, we are able to define and execute time- and event-based schedules for individual reports. In the future, we will add the ability to define schedules for administrative requests and continue to flesh out the functionality.
- *Failover.* Yi Du has been able to demonstrate failover of the Castor Server in an MSCS-based cluster. The current implementation represents the minimal level of failover support: DSS Server is treated as a general application by the clustering software.
- *Configuration Wizard.* The DSS Labs team has taken ownership of the Configuration Wizard that will be launched from the installation. Previously, we had developed a QE tool for server configuration; during the past month, this functionality has been moved to the Configuration Wizard. This tool will be incrementally enhanced between now and Alpha 3.

#### **Quality initiatives**

- *Platform testing plan.* We want to aggressively expand the number of different RDBMS platforms used in our development and testing. Over the past month, Jianhua Wang and David Weld established an infrastructure for different platform environments so that backend developers can each use a different RDBMS for their everyday work. At least 50% engineers are using the different platforms to develop, which means we have increased our platform testing coverage of the product.
- *Alpha 1 test at NDC.* Jianhua Wang participated in the last Alpha 1 site visit at NDC in Phoenix this month.
- *Expanded memory leak coverage.* Abdel Ghalayini has enhanced our automated tests for monitoring memory leaks by adding tests for a wider variety of reports, element browsing, object browsing, and client connections. As a result, we have been able to identify leaks outside of report execution. Sunil Dixit has been focused on helping to track down these leaks.

#### **Performance infrastructure**

- *Weekly performance scorecard.* During the past month, we completed our Performance Roadmap document that highlights our strategy for improving performance, defining publishable benchmarks, and developing sizing and configuration guidelines for the Server. The first step that we have taken is to publish a weekly performance scorecard that shows report execution throughput under a number of standard scenarios. Currently, these represent basic “baseline” scenarios and will be expanded to include more real-life scenarios soon.

## *Work In Progress*

### **Feature development towards Alpha 2**

- *Job Priority and Servicing.* Although we have completed the GUI front end to define report job priority and servicing schemes, we are still completing some work on the backend. For Alpha 2, we will be able to prioritize jobs by project, user, and report cost.

### **Feature testing for Alpha 2 features**

Quality Engineering is focusing on the following features:

- Scheduling.
- Failover.
- File-based report caching.
- Report views and subsetting.

### **Quality initiatives**

- *International environment certification for Alpha 2.* Abdel Ghalayini is executing certification test suites for the Server in German and Korean environments.
- *Performance analysis project.* We are developing a simple DSS project in the spirit of Warehouse Monitor that will allow us to use DSS Agent to analyze performance statistics generated by the Server.

### **Performance infrastructure**

- *New multi-threaded performance test program.* Zheng Wang is developing a new client executable that can be used with our Server Blaster program. We are already using a client executable designed for stress testing, but the new executable can be plugged-in specifically for performance testing. The chief enhancements are the ability to control the job submission rate and the ability to submit object browse and element browse requests. This new tool is important for running real-world performance scenarios because it allows us to control the distribution of report requests and the request profile of each scenario.

## *Next Steps*

- *Preparation for Alpha 2 ZDB.* As we wrap up the feature development for Alpha 2, we will turn our attention full-time towards closing down issues in TQMS and generally stabilizing the build.
- *Planning for Alpha 3.* Over the next month, we expect to begin planning for the Alpha 3 development cycle. Our challenge is to ensure that we can really be feature complete for this milestone.
- *Designs for Alpha 3.* As a result of the planning process, we will identify key design issues that need to be resolved prior to Alpha 3. As some members of the team are focused on build quality, senior engineers will get a head start on addressing design issues for Alpha 3.

## *Issues*

- *Reaching feature completeness.* A number of features have been implemented to the “stage 1” level. That is, essential functionality has been put in place so that we can perform meaningful testing and demonstrate proof of concept. It is tempting to check off these features as “complete” because the basic functionality is there, but in reality, there is quite a bit of work to do before we are feature complete.
- *Clustering development environment.* Over the past month, we had considerable difficulty resolving configuration issues with our Dell/MSCS cluster environment. We eventually had success with support from IS and our own development team, but we are not convinced that this is the right support infrastructure for clustering. Clustering environments are specialized configurations similar to the RDBMS platforms and we should support them through DSS Labs or similar means.
- *Clustering strategy.* Some recent observations have forced us to reevaluate our strategy for clustering. Note that clustering technology provides both high availability and distributed processing capability. Conventional wisdom and our own experience with our MSCS environment have prompted us to question how much we should rely on MSCS in our clustering strategy. Over the next month, we expect to refine our clustering strategy for failover and load balancing.

## ***Resources***

The resource roster for the Castor Server team is shown below.

Name	Role	Concentration	Allocation
Wayne Li	Engineering manager		FT
Kevin Wei	Engineering	Function plug-in	FT
Ningning Liu	Engineering	Caching and subsetting, report execution	FT
Janaki Goteti	Engineering	Scheduling, database connection	FT
Sunil Dixit	Engineering	Stability, scalability analysis	FT
Ramprasand Polana	Engineering lead	Scheduling, Inbox, Kernel	FT
Yi Du	Engineering	Failover	PT
Ashish Soni	Quality engineering lead	All backend teams	PT
Jianhua Wang	Quality engineering lead	Kernel	FT
Abdel Ghalayini	Quality engineering	Kernel	FT
David Weld	Quality engineering	Kernel	FT
Zheng Wang	Software test engineer		FT
Randy Hechinger	Documentation		PT
Scott Cappiello	Program management		FT

## ***Quality Report***

Current issues meeting ZDB criteria for Alpha 2

TQMS Status	Issues
TBC by QE	0
TBA	11
Preassigned	10
Assigned	98
Unfixed	0
Ready To Test	8

## ***Attachments***

DSS Server Performance Roadmap

## ***Castor Server Status: Kernel and Server Administrator***

*February 2, 1999*

### ***Summary***

Over the past month, the entire Castor team has been focused on development of features for our Alpha 2 milestone. We have formed a number of feature teams that cross the familiar product-oriented teams in order to concentrate on delivery of the *feature*. The Kernel team had an active role in new features including report caching, scheduling, report subsetting, job prioritization and servicing, inbox, and report views. While completing this backend functionality, the Server also worked with the GUI teams to complete frontend work for prioritization and servicing, report schedule creation, schedule events creation, and diagnostics configuration.

In addition to work related to these features, our quality engineering efforts have been focused on controls to help us monitor memory leaks, performance, and platform compatibility of the Castor Server. Many people will be interested in our performance analysis to date. In these tests, we are currently focusing on minimizing the overhead introduced by our architecture. We use a weekly scorecard to monitor report job throughput at a high level. This scorecard pulls all external factors out of the equation (e.g. database execution time) by running reports that return very quickly. The maximum throughput that we are achieving is around 375 report jobs per minute; which translates into about half a million jobs per day on a single server. With the current implementation of caching, we can achieve around 500 report jobs per minute. The machine we use for testing is a 4-proc box with fairly slow processors (~200 MHz). As we finalize the features for Alpha 2, we are expanding the scenarios for performance testing to generate numbers that reflect real-world usage (i.e. mixture of report complexity, cache hit ratios, non-report activity, etc.).

Also, we have run some "endurance" tests on the same configuration in which we have pushed through over two million simple report executions through the server over a period of a few days. Again, these results are achieved under simplified scenarios that do not necessarily reflect real-world conditions, but the results are encouraging nonetheless. We are beginning to step up the degree of difficulty for the server in these tests, including running million-job endurance tests that include metric qualifications and ranking reports and running result sets with more than one million rows returned.

### ***Demonstration Topics***

TBD

### ***Milestone Schedule***

Milestone	Description	Target Date
Alpha 2 ZDB	Quality milestone to coincide with Q1 Company Day. This ZDB will be taken to a set of customers for external testing.	1/22/1998
Alpha 3 ZDB	Feature complete for Server, API, and design tools. This ZDB will be available for customers for external testing.	4/2/1998

## ***Status***

### ***Achievements of the past month***

#### **Feature development towards Alpha 2**

- *Inbox.* Sergio Trejo and Ramprasad Polana completed implementation of core Inbox functionality. Using the Inbox, if a user submits a report that completes while the user is disconnected, a message appears in the Inbox to let the user know that the results are available. When combined with report caching, the user is able to retrieve results directly from the Inbox message.
- *Report caching.* Ningning Liu has been focused on using caching in the report execution cycle. Report caches allow result sets to be saved in memory and to disk so those queries do not have to be re-executed against the warehouse database. For Alpha 2, we have focused on adding file-based caches.
- *Report subsetting.* Ningning Liu has also been working on subsetting functionality, along with members of the Engine and COM teams. Subsetting allows reports to use partial results of existing caches. This will be an important feature for Broadcaster personalization. For Alpha 2, the Server supports template subsetting. Filter subsetting will be added later.
- *Report views.* The Castor Server also supports multiple simultaneous views of the same report. This will allow a report viewer to display the same result set in multiple ways at the same time (e.g. a grid and a graph, for instance).
- *Scheduling.* Andres Paz, Janaki Goteti, and Ramprasad Polana have put in place the first iteration of the Castor Server scheduling functionality. For Alpha 2, we are able to define and execute time- and event-based schedules for individual reports. In the future, we will add the ability to define schedules for administrative requests and continue to flesh out the functionality.
- *Failover.* Yi Du has been able to demonstrate failover of the Castor Server in an MSCS-based cluster. The current implementation represents the minimal level of failover support: DSS Server is treated as a general application by the clustering software.
- *Configuration Wizard.* The DSS Labs team has taken ownership of the Configuration Wizard that will be launched from the installation. Previously, we had developed a QE tool for server configuration; during the past month, this functionality has been moved to the Configuration Wizard. This tool will be incrementally enhanced between now and Alpha 3.

#### **Quality initiatives**

- *Platform testing plan.* We want to aggressively expand the number of different RDBMS platforms used in our development and testing. Over the past month, Jianhua Wang and David Weld established an infrastructure for different platform environments so that backend developers can each use a different RDBMS for their everyday work. At least 50% engineers are using the different platforms to develop, which means we have increased our platform testing coverage of the product.
- *Alpha 1 test at NDC.* Jianhua Wang participated in the last Alpha 1 site visit at NDC in Phoenix this month.
- *Expanded memory leak coverage.* Abdel Ghalayini has enhanced our automated tests for monitoring memory leaks by adding tests for a wider variety of reports, element browsing, object browsing, and client connections. As a result, we have been able to identify leaks outside of report execution. Sunil Dixit has been focused on helping to track down these leaks.

#### **Performance infrastructure**

- *Weekly performance scorecard.* During the past month, we completed our Performance Roadmap document that highlights our strategy for improving performance, defining publishable benchmarks, and developing sizing and configuration guidelines for the Server. The first step that we have taken is to publish a weekly performance scorecard that shows report execution throughput under a number of standard scenarios. Currently, these represent basic “baseline” scenarios and will be expanded to include more real-life scenarios soon.

## *Work In Progress*

### **Feature development towards Alpha 2**

- *Job Priority and Servicing.* Although we have completed the GUI front end to define report job priority and servicing schemes, we are still completing some work on the backend. For Alpha 2, we will be able to prioritize jobs by project, user, and report cost.

### **Feature testing for Alpha 2 features**

Quality Engineering is focusing on the following features:

- Scheduling.
- Failover.
- File-based report caching.
- Report views and subsetting.

### **Quality initiatives**

- *International environment certification for Alpha 2.* Abdel Ghalayini is executing certification test suites for the Server in German and Korean environments.
- *Performance analysis project.* We are developing a simple DSS project in the spirit of Warehouse Monitor that will allow us to use DSS Agent to analyze performance statistics generated by the Server.

### **Performance infrastructure**

- *New multi-threaded performance test program.* Zheng Wang is developing a new client executable that can be used with our Server Blaster program. We are already using a client executable designed for stress testing, but the new executable can be plugged-in specifically for performance testing. The chief enhancements are the ability to control the job submission rate and the ability to submit object browse and element browse requests. This new tool is important for running real-world performance scenarios because it allows us to control the distribution of report requests and the request profile of each scenario.

## *Next Steps*

- *Preparation for Alpha 2 ZDB.* As we wrap up the feature development for Alpha 2, we will turn our attention full-time towards closing down issues in TQMS and generally stabilizing the build.
- *Planning for Alpha 3.* Over the next month, we expect to begin planning for the Alpha 3 development cycle. Our challenge is to ensure that we can really be feature complete for this milestone.
- *Designs for Alpha 3.* As a result of the planning process, we will identify keys design issues that need to be resolved prior to Alpha 3. As some members of the team are focused on build quality, senior engineers will get a head start on addressing design issues for Alpha 3.

## *Issues*

- *Reaching feature completeness.* A number of features have been implemented to the “stage 1” level. That is, essential functionality has been put in place so that we can perform meaningful testing and demonstrate proof of concept. It is tempting to check off these features as “complete” because the basic functionality is there, but in reality, there is quite a bit of work to do before we are feature complete.
- *Clustering development environment.* Over the past month, we had considerable difficulty resolving configuration issues with our Dell/MSCS cluster environment. We eventually had success with support from IS and our own development team, but we are not convinced that this is the right support infrastructure for clustering. Clustering environments are specialized configurations similar to the RDBMS platforms and we should support them through DSS Labs or similar means.
- *Clustering strategy.* Some recent observations have forced us to reevaluate our strategy for clustering. Note that clustering technology provides both high availability and distributed processing capability. Conventional wisdom and our own experience with our MSCS environment have prompted us to question how much we should rely on MSCS in our clustering strategy. Over the next month, we expect to refine our clustering strategy for failover and load balancing.



## ***Resources***

The resource roster for the Castor Server team is shown below.

Name	Role	Concentration	Allocation
Wayne Li	Engineering manager		FT
Kevin Wei	Engineering	Function plug-in	FT
Ningning Liu	Engineering	Caching and subsetting, report execution	FT
Janaki Goteti	Engineering	Scheduling, database connection	FT
Sunil Dixit	Engineering	Stability, scalability analysis	FT
Ramprasand Polana	Engineering lead	Stability, Kernel internals	FT
Yi Du	Engineering	Failover / clustering	PT
Ashish Soni	Quality engineering lead	All backend teams	PT
Jianhua Wang	Quality engineering lead	Kernel	FT
Abdel Ghalayini	Quality engineering	Kernel	FT
David Weld	Quality engineering	Kernel	FT
Zheng Wang	Software test engineering	Performance	FT
Nick Pratt	Software test engineering		
Randy Hechinger	Documentation		PT
Scott Cappiello	Program management		FT

## ***Quality Report***

Current issues meeting ZDB criteria for Alpha 2

TQMS Status	Issues
TBC by QE	0
TBA	11
Preassigned	10
Assigned	98
Unfixed	0
Ready To Test	8

## ***Attachments***

DSS Server Performance Roadmap

# **Castor Server Status**

February 25, 1999

## ***Summary***

Over the past four weeks, the Castor Server team has prepared for the Alpha 3 development cycle and begun execution of the plan for this next milestone. A great deal of effort has been placed on contributing accurate engineering estimates to the cross-team project plan to ensure that we have an achievable amount of scope for the timeframes we have selected. The goal for the Kernel team in particular is to complete all Phase I feature implementation in the next 6-8 week cycle so that we can turn our full attention to stability and performance. In fact, the Kernel engineering team has shifted the organization to allow some developers to focus full-time on non-feature work for the entire cycle.

One of the most significant implementation efforts for the Alpha 3 cycle will be the integration of the Web product. Under the Castor Web architecture, a great deal of work actually occurs within the DSS Server. We have worked with the Web team to share resources, which will let us grow server-side knowledge in engineers that will become the core of the Web development effort.

On the quality front, we have identified some difficult issues affecting the overall performance and stability of the server under high concurrent usage. We have made adjustments in the team organization and developed a task list during the A3 cycle that will help us resolve this set of issues. Meanwhile, the QE team is making good progress on feature test suites for expected features in Alpha 3 and beginning to execute those tests as features become available. The QE team is also putting effort into enhancing the testing infrastructure by rotating the RDBMS platform used by the 7x24 test server, converting our internal environment to take advantage of the new NT trusted security mode, and enhancing daily memory usage and performance tests.

## ***Demonstration Topics***

Schedule Wizard

## ***Milestone Schedule***

Milestone	Description	Target Date
Alpha 3 ZDB	Feature complete for Server and API. This ZDB will be available for customers for external testing.	4/30/1998

## ***Status***

### ***Achievements of the past month***

#### **Planning for Alpha 3**

- Detailed engineering plans for Kernel engineers.

#### **Designs for Alpha 3**

- Impact of web architecture in server.
- Document object.
- Session manager.
- Functional specs for Server Admin, Scheduling, Prioritization, Cache Administration.

#### **Feature development towards Alpha 3**

- *Security control on server operations.* Implemented security checks in the server to ensure that users only perform the operations they are supposed to.
- *Scheduling.* Completed monitoring and admin interface for scheduling.
- *Server Admin.* Enhancements to schedule manager and connection mapping interface.

### **Quality initiatives**

- *Finalized Alpha 2 development cycle.* Quite a bit of time was spent this month bringing the Alpha 2 development cycle to a close. The team as a whole hit a point of diminishing returns and ended up frustrated by not achieving our objectives.
- *Alpha 2 visit to Glaxo* Abdel Ghalayini and Pat Orie participated in the Alpha site visit to Glaxo this month.
- *Transitioned daily memory leak monitoring to build process.* This will let us catch new memory leaks as quickly as possible.

### **Performance infrastructure**

- *Daily performance tests.* We continued to execute performance tests as a way to monitor the daily build. These tests have been severely hampered by a set of bugs that are still lingering.
- *Stress testing.* We conducted all-hands stress testing as well as more structured stress testing of the server. These tests did not progress very far for the same issues uncovered by performance testing.

### **Migration efforts**

- *Alpha 2 visit to Western Digital.* Pat Orie participated in the Alpha site visit to Western Digital this month.
- *Migration issue list.* Began a list of frequently-asked questions and significant issues for customers facing a migration to the Castor platform. This will be a living document as we conduct more Alpha sites.

## **Work In Progress**

### **Feature development towards Alpha 3**

- *Job Priority and Servicing.* Enhancements to cost-based priority, priority by user groups.
- *Server Admin.* Diagnostics enhancements, importing users from NT, security access to application functionality, VLDB properties editor.
- *Web support.* Continue work on Web API and server support for interpreting XML-based requests.
- *Lock stack manager.* Enhancements to our lock stack manager infrastructure will help us track down and even prevent the deadlock issues that are at the root of our performance problems.
- *Prompt caching.* Enhancements to report caching to resolve prompts than can access caches.

### **Feature testing for Alpha 3 features**

- Trusted security and access control on server operations.
- Schedule monitoring.

### **Quality initiatives**

- *Platform testing.* Our 7x24 machine for running Castor Server has been configured to rotate database platform on a weekly basis. This will give us additional coverage for our platform testing "for free," as everyone exercises a wide variety of features against this machine.
- *Test suites for Alpha 3 features.* We are off to a good start with the test suites for features that are coming during the next 6-8 weeks.
- *Trusted security mode.* Dominique is working on configuring the 7x24 environment to operate using trusted security mode. This will let us operate in an more client-realistic environment, and flesh out corresponding issues.
- *Usability testing.* The QE Usability team is providing feedback on the Server Admin interface. Feedback on this functionality has been quite sparse to date.

## **Performance infrastructure**

- *Warehouse monitor project.* We have begun the development of a Castor project that runs against the Castor statistics tables. We expect to derive multiple benefits from this effort: 1) we will have a convenient way to analyze our own performance data, 2) we will refine our statistics module based on our own analytical requirements, 3) we will identify any bugs in important paths of the statistics code, and 4) we will create the predecessor of a Warehouse Monitor that we can ultimately package as a product.
- *Moving performance tests to a more complicated project.* We have had some trouble setting up a copy of the Glaxo project in our performance lab. We are still working through this, as it will allow us to create more complex usage scenarios and gather more interesting performance numbers than the baseline numbers we are getting right now.

## ***Issues***

- *Performance and stability issues.* The issues hampering our performance and stability tests mentioned above are very serious for the health of the product. Excerpts from a recent email describes our approach:
  - To address these issues, we are planning tasks during Alpha 3 to ensure that all code modules are taking advantage of kernel infrastructure that can help us better troubleshoot concurrency issues when they occur. We will also make enhancements to the Server's Lock Stack Manager, which will allow us to detect concurrency issues without having to actually experience a problem.
  - We will also attack these issues via code review. Wayne Li is leading the effort on a list of specific (non-GUI) code reviews.
  - For the Alpha 3 development cycle, our plan calls for a separate team of developers who will focus on performance and stability (no feature development). This team currently consists of Sunil Dixit, with plan to grow after the T2 bootcamp.
  - The performance infrastructure team formed during Alpha 2 will continue to develop tools, test scripts, etc. for the purposes of assessing performance. This team currently consists of Zheng Wang, with plan to grow after the T1 bootcamp.
  - We have moved to daily performance monitoring (instead of weekly) so that we can catch issues as soon as they are introduced, potentially even as part of build acceptance. We expect that we will receive feedback from the entire team on how to improve these tests so that the information we monitor is actionable. Ashish Soni is currently broadcasting this information and will soon coordinate the effort with the QE Enterprise Systems Analysis group.
  - Some consolation is the fact that we have achieved better performance results through the Castor Server in previous builds (baseline report throughput in the range of 350-400 reports per minute). To some extent, we have taken our eye off the ball in letting performance slip to the levels reported this week. The action items described above will let us recover to previous levels and then begin the process of optimizing to reach the throughput expected of the Castor Server.
- *Migration to Clearcase.* The migration to Clearcase has been somewhat rocky. The critical aspect is that the daily build has not been consistent over several days. We are increasing the risk of not catching memory leak or performance issues introduced each day.
- *Impact of Web effort.* The majority of Kernel implementation work during the Alpha 3 development cycle is related to the Web. We are aggressively adjusting scope to be able to serve the Web as a priority and still be feature complete for Alpha 3.

## ***Resources***

The resource roster for the Castor Server team is shown below.

<b>Name</b>	<b>Position</b>	<b>Team</b>	<b>Notes</b>
<b><i>Kernel Engineering</i></b>			
Wayne Li	Engineering Manager		
Ramprasad Polana	Lead Engineer	System Debug Team	Implementation up to 3/6; vacation 3/8 – 4/2
Nick Pratt	Engineer	System Debug Team	
Sunil Dixit	Lead Engineer	System Diagnostics Team, Stability Team	
Parker Zhang	Engineer	Stability Team	Web integration up to 4/15
Zheng Wang	Test Engineer	Performance Team	
Tina Tian	Test Engineer	Performance Team	
Ningning Liu	Engineer	Report Execution	
Yi Du	Engineer	Clustering	
Kevin Wei	Engineer	Web API, Clustering	
Janaki Goteti	Engineer	Web Integration	Vacation 2/22 – 3/12
Sam Helwig	Engineer	Web Integration	On loan from Castor web team
Ping Xu	Engineer	Kernel; Web Integration	Implementation prior to T5 bootcamp
<b><i>Server QE</i></b>			
Ashish Soni	Quality engineering lead	All backend teams	
Jianhua Wang	Quality engineering lead	Kernel	
Abdel Ghalayini	Quality engineering	Kernel	
David Weld	Quality engineering	Kernel	
Dominique Paschoud	Quality engineering	Kernel	
<b><i>Documentation</i></b>			
Randy Hechinger	Tech Writer		
<b><i>Programs</i></b>			
Scott Cappiello	Program Manager		
Pat Orié	Programs Engineer	Migration	

## ***Quality Report***

Current issues meeting ZDB criteria for Alpha 3.

<b>TQMS Status</b>	<b>Issues</b>
TBC by QE	1
TBA	18
Preassigned	5
Assigned	53
Unfixed	0
Ready To Test	3
Postponed	99

## ***Attachments***

Feature Release Plan

## ***Castor Server Status***

*March 26, 1999*

### ***Summary***

During the past month, the Server team has continued work on the feature set for the Alpha 3 milestone. Features in progress include support for the Web version of the Inbox, enhancements to the report execution cycle, progress on the Castor Web API, and basic clustering membership. New engineers joining the team are making contributions to the web support effort as well as in areas such as report caching. We have also introduced two new test engineers to the team who will make an immediate impact in the areas of stability and performance.

On the quality front, we have been able to give testing coverage to some features that have come through in the daily build. In the past month, we have made progress on security access control on server operations, trusted security mode, scheduling, and job prioritization. Also, the Usability team executed usability studies on server-related user interfaces including Server Administrator and the Configuration Wizard.

One of the challenges we are facing in this stage of the product's development involves a set of stability issues that the server experiences under high user concurrency. In the past month, we completed one part of our action plan, which included enhancing the Server Kernel's infrastructure for detecting potential problems. The second part is still under way as other code modules are updated to take advantage of this infrastructure. It is very important that we continue to give this effort very high priority. The benefits should be great as we will be able to detect in our own labs the sort of bizarre issues that in the past have led to customer fires and site visits for our engineers.

Another area where we have made progress is in focusing on the Castor Migration Experience. We should all recognize that the Castor architecture is significantly different from the current product architecture and that existing projects will undergo some important changes as they move into the Castor world. The goal for our Migration team is to ensure that customers can make the smoothest possible transition to Castor.

Based on our Alpha site visits, we have begun to compile a list of issues and frequently asked questions related to migration. On the product side, this helps us double-check our support for 100% of major 5.x functionality requirements and meeting major 6.0 enhancement requests. On the service side, this information will grow into a project methodology that tells customers/partners/consultants how they can migrate their current projects, including tips for dealing with workarounds in the old architecture. We expect this effort to continue to ramp up as we move through Alpha and Beta cycles.

### ***Demonstration Topics***

Schedule Wizard and Schedule Administration

### ***Milestone Schedule***

Milestone	Description	Target Date
Alpha 3 Backend Code Freeze	Code freeze for Alpha 3 feature set.	4/30/1999
Alpha 3 ZDB	Feature complete for Castor Phase 1. This ZDB will be available for customers for external testing.	5/15/1999

## ***Status***

### ***Achievements of the past month***

#### **Planning for Alpha 3**

- Wrapped up final scope for Castor Phase 1.

#### **Designs for Alpha 3**

- Updated functional specs for Configuration Wizard, Clustering.

#### **Feature development towards Alpha 3**

- *Server Admin.* Diagnostics enhancements, importing users from NT, security access to application functionality, VLDB properties editor, schedule monitoring.
- *Lock stack manager.* Enhancements to our lock stack manager infrastructure will help us track down and even prevent the deadlock issues that are at the root of our performance problems.
- *Clustering.* Basic cluster manager is in place, which allows servers to join and leave a cluster.
- *Web support.* Revised Web API and implemented server support for authentication, browsing objects, executing reports, browsing elements. Introduced Document definition object into backend.
- *Enhancements to report execution cycle.* Backend support for various improvements in report execution.

#### **Quality initiatives**

- *Usability testing.* The QE Usability team is providing feedback on the Server Admin interface. Feedback on this functionality has been quite sparse to date.
- *Feature testing.* Access control on server operations, trusted security mode, scheduling, prioritization, cluster membership.
- *Platform testing.* The 24x7 database rotation gave us coverage of Oracle 7.3, DB2/UDB, and Informix ODS.

#### **Performance infrastructure**

- *Enterprise Systems Analysis.* Enterprise systems analysis team has taken responsibility for running daily monitoring tests.

#### **Migration efforts**

- *Migration FAQ.* Completed the first iteration of the Castor migration frequently-asked questions list.
- *Site visit to Allegheny Ludlum Corporation.* Pat Orie and Olivier Marchal conducted an Alpha site visit to ALC.

### ***Work In Progress***

#### **Feature development towards Alpha 3**

- *Server Admin.* Changes to Database Instance editor, ability to import VLDB drivers, cluster administration.
- *Web support.* Continue work on Web API and server support for interpreting XML-based requests.

#### **Feature testing for Alpha 3 features**

- *Job Priority and Servicing.* Preliminary testing while waiting for Ramp to return from India.
- *Scheduling.* Preliminary testing while waiting for Ramp to return from India.

#### **Quality initiatives**

- *Platform testing.* Our 7x24 machine for running Castor Server has been configured to rotate database platform on a weekly basis. This will give us additional coverage for our platform testing "for free," as everyone exercises a wide variety of features against this machine.
- *Alpha 2 visit to Payless.* David Weld and Olivia Moncayo are on site at Payless ShoeSource.

- *Feature testing.* Report execution enhancements.

#### **Performance infrastructure**

- *Warehouse monitor project.* We have begun the development of a Castor project that runs against the Castor statistics tables. We expect to derive multiple benefits from this effort:
  - 1) we will have a convenient way to analyze our own performance data,
  - 2) we will refine our statistics module based on our own analytical requirements,
  - 3) we will gain insight into the end-to-end process of building and managing a Castor project, and
  - 4) we will create the predecessor of a Warehouse Monitor that we can ultimately package as a product.
- *Moving performance tests to a more complicated project.* We have converted the 5.x project from Premier into a Castor project in our performance lab. We are still working through the creation of more complex usage scenarios. This will allow us to gather more interesting performance numbers than the baseline numbers we are getting right now.

#### ***Issues***

- *Open stability issues.* We are still working on a set of stability issues that the server experiences under moderate user concurrency. In the past month, we completed one part of our recovery plan, which included the enhancements to the Server's Lock Stack Manager to help us detect potential cycles. The second part is still under way as changes in COM modules are implemented to take advantage of centralized locking infrastructure. It is very important that we continue to give this effort very high priority.
- *Stalled performance effort.* Because of the stability issues above and our focus on planning and managing the remaining features that comprise minimum scope of the product, we are not giving enough attention to performance. The Enterprise Systems Analysis team and other QE teams have made resources available for running tests and monitoring performance



## Resources

The resource roster for the Castor Server team is shown below.

Name	Role	Team	Notes
<b>Kernel Engineering</b>			
Wayne Li	Engineering Manager		
Ramprasad Polana	Software Engineering	System Debug Team	Vacation 3/8 – 4/2
Nick Pratt	Software Engineering	System Debug Team	
Sunil Dixit	Software Engineering	System Diagnostics Team, Stability Team	
Zheng Wang	Software Test Engineering	Stability and Performance	
Yonghui "Huge" Wang	Software Test Engineering	Stability and Performance	
Lixin Li	Software Test Engineering	Stability and Performance	
Ningning Liu	Software Engineering	Report Execution, Web Module	
Yuxiao Xiao	Software Engineering	Report Execution, Web Module	
Tina Tian	Software Engineering	Report Execution	
Yi Du	Software Engineering	Clustering	
Kevin Wei	Software Engineering	Web Module, Clustering	
Janaki Goteti	Software Engineering	Web Module; Session Manager	
Yuan Ding	Software Engineering	Web Module	On loan from Web
Sam Helwig	Software Engineering	Web Module	On loan from Web
Ping Xu	Software Engineering	Session Manager	Implementation prior to T5 bootcamp
<b>Server QE</b>			
Ashish Soni	Quality Engineering	Lead for all backend teams	Kernel QE 50%
Jianhua Wang	Quality Engineering	Lead for Kernel QE	
Abdel Ghalayini	Quality Engineering	Kernel QE	Vacation until 4/16
David Weld	Quality Engineering	Kernel QE	
Dominique Paschoud	Quality Engineering	Kernel QE	
<b>Documentation</b>			
Randy Hechinger	Tech Writer		
<b>Programs</b>			
Scott Cappiello	Program Manager		
Pat Orie	Programs Engineer	Castor Migration	

## Quality Report

Current issues meeting ZDB criteria for Alpha 3.

TQMS Status	Issues
TBC by QE	2
TBA	29
Preassigned	20
Assigned	79
Unfixed	1
Ready To Test	4
Postponed	23
<b>TOTAL</b>	<b>158</b>

## Attachments

Cross-team Development Plan  
Feature Release Plan

## ***Castor Server Status***

*May 3, 1999*

### ***Summary***

During the past month, the Server team has continued work towards the Alpha 3 milestone. Recent achievements include Web XML API support for the Inbox, as well as continued support for basic operations such as authentication, element browsing, object browsing, and report execution. Also, the engineering team has enhanced the caching feature set to include file-based caches and administration and monitoring capabilities. Finally, a number of new features are now available in the Server Administration GUI, reflecting the new functionality of the backend and addressing feedback from the Usability team.

As mentioned last month, we have dedicated team members working in the areas of stability and performance. This group is responsible for improving memory usage and leakage in the server, identifying areas for performance optimization, maintaining a usable and efficient diagnostics infrastructure, and otherwise enhancing the stability and scalability of the server. In the past month, this team has helped to lift overall system performance to more acceptable levels and begun the long road towards making sure the server is bulletproof. Realistically, this is probably the most daunting challenge facing the team. We know that the Castor product suite is considerably more complex than the existing architecture, and we have our work cut out for us to make sure that the next generation of software is as ready for customers as the current one.

At the same time, quality engineers have been giving test coverage to the web backend, the new session manager and enhanced inbox, basic clustering and load balancing capability, and the report execution cycle. Server QE has also expanded platform coverage for major database platforms, including support for Oracle as a metadata platform. In the coming month, quality engineering will be focused on ensuring that the Alpha 3 build is ready for customer consumption. In addition, quality engineers are preparing for product knowledge transfer that will be essential as we move towards Beta; members of the Tech Support team are rotating through as "guest QE's" in the next few weeks.

A new effort that began this month was the creation of a Castor Warehouse Monitor team. Dave Hutz and Sascha Naujoks are responsible for requirements analysis and the delivery plan for a Castor Warehouse Monitor product. While the Castor Server already features a statistics module and Server QE has been maintaining a warehouse monitor project for internal use, the new Warehouse Monitor team is ready to take this to the next level. We look forward to rapid progress in the coming weeks.

### ***Demonstration Topics***

VLDB driver upgrade

### ***Milestone Schedule***

Milestone	Description	Target Date
Alpha 3 Backend Feature Freeze	Freeze on all feature development for backend teams.	4/30/1999
Alpha 3 Feature Freeze	Code freeze for Alpha 3 feature set.	5/14/1999
Alpha 3 ZDB	Feature complete for Castor Phase 1. This ZDB will be available for customers for external testing.	5/28/1999
Beta 1 ZDB target	Cleanup tasks complete and software is of sufficient quality for an MSI Way Beta.	7/3/1999

## ***Status***

### ***Recent Progress***

#### **Feature Implementation**

- *Web XML API.* Continued and enhanced support for basic operations (authentication, element browsing, object browsing, report execution, report sorting, report pivoting, report page-by and prompts). Completed integration of the Web Inbox. More details are available in the Web report.
- *Cache administration.* Added the ability to monitor and manipulate report caches that exist in server memory and on disk. Completed the implementation of file-based caches and swapping logic.
- *Document processing.* We are in the process of integrating the document object into the server for the first time.
- *Graph processing for Web.* We have recently integrated the ability to create graphs on the server for display via the Web.
- *Clustering.* Clustered servers now support metadata synchronization, ensuring that metadata objects that are changed on one server can be propagated to other servers. Also, user sessions submitted through the Web are load balanced according to which server have the fewest open sessions.
- *Server Admin.* New features include the database instance wizard, usability enhancements, cache monitoring and administration, cluster administration, and the multidimensional security editor.

#### **Stability and Performance**

- *Performance enhancements.* Identified frequently used code in our diagnostics infrastructure, and made enhancements in those modules to improve overall system throughput. Also, the Metadata Server team optimized the number of SQL statements that were being used and greatly enhanced object browsing.
- *Diagnostics.* We have implemented a number of changes to make our diagnostics output more readable. This will help us more easily identify unnecessary function calls and other inefficiencies just by looking at ordinary diagnostics output. Mala Viswanath is focused on analyzing this output and building a list of potential inefficiencies.
- *Memory leak analysis.* We are currently focused on memory leaks for basic server operations. We have not driven down leaks to zero for all of these operations, although we are making progress on server startup/shutdown and report execution for simple reports.
- *Deadlock analysis.* We have completed the infrastructure necessary to help us find potential deadlocks in the system and cleaned up a number of small deadlocks. Unfortunately, we are still plagued some known deadlocks that have proven difficult to track down and eliminate.

#### **Quality Initiatives**

- *Feature testing.* Report execution enhancements, cluster membership, backend support for web, session manager and inbox enhancements, XML API test programs.
- *Alpha 2 visit to Payless.* David Weld and Olivia Moncayo completed a site visit to Payless ShoeSource in the beginning of April.
- *Platform testing.* The 24x7 database rotation program has given us coverage of all databases we plan to support except for Tandem and DB2/390. Also, we have tested Oracle as a metadata platform and included this in the rotation program.
- *StockMarket project.* We have migrated the StockMarket project used by Telepath to the Castor Environment and identified a subset of reports that run correctly.

#### **Performance Analysis**

- *Daily monitoring.* Enterprise Systems Analysis team has taken responsibility for running daily monitoring tests.
- *Alpha 3 objectives.* We are measuring two sets of metrics for Alpha 3. The first is a series of maximum throughput tests for standard server operations, such as element browsing, object browsing, and report execution (3- and 4-tier). The second is a series of user concurrency measurements for the same operations.

- *Standard benchmarks.* The Performance Analysis team is defining customer-based scenarios to ensure that our platform can scale to meet customer requirements.

#### **Migration**

- *Migration FAQ.* Continued development and maintenance of frequently asked questions list based on nine Alpha site visits, reviews with Tech Support, Consulting, and customer meetings.
- *Castor knowledge base.* Began development of knowledge base for Castor.
- *Castor Upgrade manual.* Continued to provide material for Upgrade documentation.

#### **Warehouse Monitor**

- *Formed a WH Monitor team.* David Hutz is driving the requirements for Castor Warehouse Monitor and directing the execution of the delivery plan. Sascha Naujoks is responsible for WH Monitor development.
- *Reviewed existing warehouse monitor project.* Dominique Paschoud has turned over the warehouse monitor project previously maintained by QE to the WH Monitor team.
- *Reviewed Castor statistics implementation.* The WH Monitor team has submitted a list of enhancement requests for the Castor statistics modules based on Warehouse Monitor requirements. This will be reviewed and addressed in May.

#### *Next Steps*

##### **Planning**

- *Beta 1.* We will begin the planning process for the Beta 1 development cycle.

##### **Feature Development**

- *Web XML API.* Continue support for drilling, NT authentication, and anonymous authentication. Support for Document Execution. Enhance Inbox functionality. Enhance support for personalization.
- *Security:* Integrate multidimensional security into report caching, enhance report subsetting to take advantage of MD security. Finish object access check in server.
- *Report Execution and Caching:* Add intelligent invalidation/update of cache contents.
- *Document Object and Execution:* Finish first end-to-end document creation and execution.
- *Clustering.* Complete testing of metadata synchronization and begin development of cache synchronization and session synchronization to support fail over.
- *Miscellaneous cleanup.* There are a number of miscellaneous feature-related tasks and enhancements that we expect to make before Alpha 3.
- *Bug fixing.* We will clear out the backlog of TQMS issues.

##### **Stability and Performance**

- *Memory leaks.* Expand analysis scope to include usage of XML API and more code paths for report execution (i.e. more complex reports).
- *Deadlocks.* Continue to trace the five known deadlocks and periodically run all-hands stress tests to reproduce deadlock errors.
- *Performance.* Continue to gather more potential optimizations based on diagnostics analysis and code review. Also, we expect a new team member to research compiler optimization and the issues that will inevitably turn up when this occurs.

##### **Feature Testing**

- Web XML API.
- Caching and cache administration.
- Document processing.
- Graph server.
- Clustering.
- Server Admin.

### Quality Initiatives

- *Platform testing.* We will tighten up the platform test suites to establish a platform certification process similar to the one we use for the current products. Also, we will continue the platform rotation program.
- *DB2 as metadata platform.* The development team expects to add support for DB2 as a metadata platform in the coming weeks. This will be incorporated into the platform testing plan.
- *APS rotation.* Several members of the Technical Support team will each spend a week on the Server QE team to transfer product knowledge.
- *Alpha 3 site visits.* The Castor QE team has scheduled site visits towards the end of May and Server QE will be involved onsite as well as providing remote support.

### Performance Analysis

- *Alpha 3 objectives.* Measure throughput and concurrency numbers to ensure that we have met our Alpha 3 targets.
- *Benchmark scenarios.* Complete the first iteration of customer scenario benchmarks and take an initial reading.

### Migration

- *Enhance existing documentation.* Continue to enhance and maintain the FAQ, migration knowledge base, and upgrade manual.

### Warehouse Monitor

- *Begin earnest development.* Develop project as new statistics features become available.

### Issues

- *Lacking Server QE resources.* The team recently lost one QE and will lose another QE resource in July. In addition, the Kernel engineering team has doubled in size in the past two months. Two Server QE resources should be added.
- *Project management support.* The server team needs a dedicated project manager to handle resource scheduling for the 12-16 developers and test engineers on the team.
- *Program management support.* The server team needs additional program management resources to focus on the following areas within the program:
  - Stability and performance.
  - Feature completion during Beta.
- *Stability and concurrency.* As mentioned in the status above, we are still plagued by deadlocks under concurrency. We have set the objective for Alpha 3 that we must be able to run an all-hands stress on the server before we will let the Alpha 3 build go to customer site.

### Resources

The resource rosters for the Castor Server team, Migration team, and Warehouse Monitor team are shown below.

Name	Role	Team	Notes
<b>Development Engineers</b>			
Wayne Li	Engineering Manager		
Ramprasad Polana	Software Engineering, team lead	System Debug Team	
Nick Pratt	Software Engineering	System Debug Team	Dead lock detection, performance
Sunil Dixit	Software Engineering, team lead	Stability and Performance Team	Memory Leak
Zheng Wang	Software Test Engineering	Stability and Performance Team	Performance/tracing
Lixin Li	Software Test Engineering	Stability and Performance Team	Memory leak/foot print
Juan Muraira	Software Engineering	Stability and Performance Team	Memory access errors

Ningning Liu	Software Engineering, team lead	Report Execution, Report Caching	
Yuxiao Xiao	Software Engineering	Report Execution, Graph Engine	
Tina Tian	Software Engineering	Report Execution, Cache admin	
Janaki Goteti	Software Engineering	Web XML API: Execution Flow	
Ping Xu	Software Engineering	Web XML API: Session Manager	T5 bootcamp
Yuan Ding	Software Engineering	Web XML API: Client component	
Yi Du	Software Engineering	Clustering	
Kevin Wei	Software Engineering	Server security	
Sam Helwig	Software Engineering	Document Object and Document Execution	
Yonghui "Huge" Wang	Software Test Engineering	Build Team	Team build
<b>Quality Engineers</b>			
Ashish Soni	Quality Engineering	Lead for all backend teams	Kernel QE 50%
Jianhua Wang	Quality Engineering	Lead for Kernel QE	
David Weld	Quality Engineering	Kernel QE	
Dominique Paschoud	Quality Engineering	Kernel QE	
<b>Documentation</b>			
Randy Hechinger	Tech Writer		
<b>Programs</b>			
Scott Cappiello	Program Manager		

Name	Role	Team	Notes
<b>Migration Team</b>			
Pat Ori	Programs Engineer	Castor Migration	

Name	Role	Team	Notes
<b>Warehouse Monitor Team</b>			
David Hutz	Program Manager	Warehouse Monitor	
Sascha Naujoks	Warehouse Monitor Engineer	Warehouse Monitor	

## Quality Report

Current issues meeting ZDB criteria for Alpha 3.

TQMS Status	Issues
TBC by QE	12
TBA	46
Assigned	141
Unfixed	2
Ready To Test	8
Postponed	23
<b>TOTAL</b>	<b>232</b>

## Attachments

Cross-team Development Plan  
Feature Release Plan  
Performance Summary

## ***Castor Server Status***

*May 27, 1999*

### ***Summary***

During the past month, the Server team has been wrapping up work for the Alpha 3 milestone. Recent achievements include Web XML API support for prompts, report manipulation, general diagnostics, and generation of graphs for the Web. A particularly exciting new addition is the first stage of the Document object, which includes basic support in the execution cycle. Documents are essentially layouts that contain multiple reports. A client can now submit a single request for a document, and that request will be broken down into constituent reports, each executed independently within the server. In addition to these efforts, the development team has turned its attention to closing down known issues with features and preparing for the Alpha 3 release.

At the same time, we have a group of developers focused on stability and performance. We have made noticeable progress in our periodic "all-hands" stress tests of the server. These tests still introduce random behavior, generating good issues that our simulated stress tests do not uncover. In the past month, we have been able to extend the length of time that the server can withstand these stresses.

Server QE has focused a lot of effort on support for the XML API, helping the Web QE team troubleshoot and also providing a layer of preliminary testing on the API in order to increase the productivity of the Web development team. We have also given feature test coverage to SQL Cancel, Prioritization, Scheduling, Caching and Cache Administration, Inbox, and Clustering. Finally, Server QE has also expanded platform coverage for major database platforms, including support for Oracle as a metadata platform.

### ***Demonstration Topics***

Document editor and document execution

### ***Milestone Schedule***

Milestone	Description	Target Date
Alpha 3 ZDB	Feature complete for Castor Phase 1. This ZDB will be available for customers for external testing.	6/3/1999
Beta 1 ZDB target	Cleanup tasks complete and software is of sufficient quality for an MSI Way Beta.	7/10/1999

### ***Status***

#### ***Recent Progress***

##### **Feature Implementation**

- *Web XML API.* Added support for prompts, report manipulation, generation of graphs for the web, as well as general diagnostics and error handling.
- *Document Object and Execution:* Finished first end-to-end document creation and execution. This is the first stage, which does not include prompts or XML API support.
- *Graph processing for Web.* Integrated the ability to create graphs on the server for display via the Web.
- *Bug Fixing:* We are earnestly fixing bugs for Alpha 3.

## **Stability and Performance**

- *Performance enhancements.* We have made some changes to improve the performance of login authentication. This should also allow us to increase concurrency in general.
- *Memory leak analysis.* For some builds, we have been able to drive our leak tests for basic operations down to zero. We are confident in the results we find from tools like Boundschecker. However, our daily tests using our in-house test programs still reveal memory consumption. We are trying to determine if the issue is with the tests or with the product.
- *Deadlock analysis.* We have removed a number of deadlocks from the system. Nonetheless, as we exercise more code paths, we discover new potential cycles as well. We expect to monitor our progress including an indication of how much code coverage we are achieving with our analysis.
- *All hands stress tests.* The efforts above have been demonstrated in the periodic all-hands stress tests run against the server. Previously, moderate concurrent activity could cause a deadlock on the server after 5-10 minutes of use. Recently, we have been able to survive hour-long stress tests without deadlocks. There are still other bugs and performance issues that are uncovered during these stress tests, but we are seeing progress.
- *Compiler optimization.* We have a dedicated person working on using compiler optimization. We expect many issues to unfold in the course of turning optimization on and Andres Murillo is responsible for driving the effort.

## **Quality Initiatives**

- *Feature testing.* We have given test coverage to all features of the XML API, SQL Cancel, Prioritization, Scheduling, Caching and Cache Administration, Inbox, Clustering.
- *Platform testing.* The 24x7 database rotation program has given us coverage of all databases we plan to support except for Tandem and DB2/390.
- *Metadata certification.* Also, we have tested Oracle 7.3 as a metadata platform and included this in the rotation program. Next steps are DB2, Oracle 8.0, and Oracle 8i.
- *Support for performance analysis.* Ashish Soni continues to support the performance analysis team representing server QE.
- *Alpha site visit to NDC.* Jianhua Wang visited NDC for an Alpha site visit.

## **Performance Analysis**

- *Daily monitoring.* Enterprise Systems Analysis team continues for running daily monitoring tests.
- *Alpha 3 throughput objectives.* We are measuring maximum throughput tests for standard server operations, such as element browsing, object browsing, and report execution (3- and 4-tier). These test results are the ones used for daily monitoring.
- *Alpha 3 concurrency objectives.* We are also measuring user concurrency for the same operations. Bugs in the product that are still under research have hindered these test results.
- *Standard benchmarks.* The Performance Analysis team is defining customer-based scenarios to ensure that our platform can scale to meet customer requirements. Once the concurrency tests are adequately passed, we expect to run a benchmark scenario based on Best Buy's usage profile.

## **Next Steps**

### **Planning**

- *Beta 1.* We will continue the planning process for the Beta 1 development cycle.

### **Feature Development**

- *Web XML API.* Add support for drilling, NT authentication, and anonymous authentication. Add support for Document execution. Enhance Inbox functionality. Enhance support for personalization. Add administrative capability through the XML API.
- *Security:* Integrate multidimensional security into report caching, enhance report subsetting to take advantage of MD security. Finish object access check in server.
- *Report Execution and Caching:* Add intelligent invalidation/update of cache contents.
- *Document Object and Execution:* Add support for prompts and graphs. Optimize for performance.
- *Clustering.* Add cache synchronization and session synchronization to support fail over.



- *Miscellaneous cleanup.* There are a number of miscellaneous feature-related tasks and enhancements that we expect to make during Beta 1.
- *Warehouse Monitor support.* Make changes to statistics tables based on Warehouse Monitor requirements.
- *Broadcaster Aurora support.* Enhance request object and other infrastructure to support Broadcaster integration.
- *Web Deuce support.* Provide translation layer so for Web Deuce product.

#### **Stability and Performance**

- *Performance.* Continue research on compiler optimization and the resolve issues that will inevitably turn up when this occurs.
- *Memory leaks.* Keep basic operations leak-free. Expand coverage to include operations through XML API and more engine features.
- *Stability and deadlocks.* Increase code coverage of deadlock analysis.

#### **Feature Testing**

- Web XML API.
- Document processing.

#### **Quality Initiatives**

- *Close out Alpha 3 milestone.* Ensure that we reach ZDB.
- *Platform testing.* We will tighten up the platform test suites to establish a platform certification process similar to the one we use for the current products. Also, we will continue the platform rotation program.
- *DB2 as metadata platform.* The development team expects to add support for DB2 as a metadata platform in the coming weeks. This will be incorporated into the platform testing plan.
- *Alpha 3 site visits.* The Castor QE team has scheduled site visits in June and Server QE will be involved onsite as well as providing remote support.

#### **Performance Analysis**

- *Alpha 3 objectives.* Measure throughput and concurrency numbers to ensure that we have met our Alpha 3 targets.
- *Benchmark scenarios.* Complete the first iteration of customer scenario benchmarks and take an initial reading.

#### ***Issues***

- *Impact of Broadcaster Aurora and Web Deuce.* In order for these development efforts to be successful, the Castor Server must make some changes for appropriate support. These efforts introduce code risk during an important part of our development cycle and also impact our resource allocation for both developers and quality engineers.
- *Project management support.* The server team needs a dedicated project manager to handle resource scheduling for the 12-16 developers and test engineers on the team.
- *Lacking Server QE resources.* The team recently lost one QE and will lose another QE resource in July. In addition, the Kernel engineering team has doubled in size in recent months. Finally, the scope of the server functionality continues to increase as we support Aurora and Deuce. Two Server QE resources should be added.

## Resources

The resource rosters for the Castor Server team are shown below.

Name	Role	Team	Notes
<b>Development Engineers</b>			
Wayne Li	Engineering Manager		
Ramprasad Polana	Software Engineering, team lead	System Debug Team	
Nick Pratt	Software Engineering	System Debug Team	Dead lock detection, performance
Sunil Dixit	Software Engineering, team lead	Stability and Performance Team	
Zheng Wang	Software Test Engineering	Stability and Performance Team	
Lixin Li	Software Test Engineering	Stability and Performance Team	
Juan Muraira	Software Engineering	Stability and Performance Team	
Andres Murillo	Software Engineering	Stability and Performance Team	
Ningning Liu	Software Engineering, team lead	Report Execution, Report Caching	
Yuxiao Xiao	Software Engineering	Report Execution, Graph Engine	
Tina Tian	Software Engineering	Report Execution, Cache admin	
Janaki Goteti	Software Engineering	Web XML API: Execution Flow	
Ping Xu	Software Engineering	Web XML API: Session Manager	
Yuan Ding	Software Engineering	Web XML API: Client component	
Yi Du	Software Engineering	Clustering	
Kevin Wei	Software Engineering	Server security	
Sam Helwig	Software Engineering	Document Object and Document Execution	
Yonghui "Huge" Wang	Software Test Engineering	Build Team	Team build
Longying Zhao	Software Engineering		
<b>Quality Engineers</b>			
Ashish Soni	Quality Engineering	Lead for all backend teams	
Jianhua Wang	Quality Engineering	Lead for Kernel QE	
David Weld	Quality Engineering	Kernel QE	
Dominique Paschoud	Quality Engineering	Kernel QE	
Elsa Polo	Quality Engineering	Kernel QE	
<b>Documentation</b>			
Randy Hechinger	Tech Writer		
<b>Programs</b>			
Scott Cappiello	Program Manager		

## Quality Report

Current issues meeting ZDB criteria for Alpha 3.

TQMS Status	Issues
TBC by QE	13
TBA	19
Assigned	134
Unfixed	0
Ready To Test	45
Postponed	0
<b>TOTAL</b>	<b>232</b>



# **Castor Server Status**

July 7, 1999

## ***Summary***

During the past month, the Server team has been wrapping up work for the Alpha 3 milestone. Recent achievements include Web XML API support for prompts, report manipulation, general diagnostics, and generation of graphs for the Web. A particularly exciting new addition is the first stage of the Document object, which includes basic support in the execution cycle. Documents are essentially layouts that contain multiple reports. A client can now submit a single request for a document, and that request will be broken down into constituent reports, each executed independently within the server. In addition to these efforts, the development team has turned its attention to closing down known issues with features and preparing for the Alpha 3 release.

At the same time, we have a group of developers focused on stability and performance. We have made noticeable progress in our periodic "all-hands" stress tests of the server. These tests still introduce random behavior, generating good issues that our simulated stress tests do not uncover. In the past month, we have been able to extend the length of time that the server can withstand these stresses.

Server QE has focused a lot of effort on support for the XML API, helping the Web QE team troubleshoot and also providing a layer of preliminary testing on the API in order to increase the productivity of the Web development team. We have also given feature test coverage to SQL Cancel, Prioritization, Scheduling, Caching and Cache Administration, Inbox, and Clustering. Finally, Server QE has also expanded platform coverage for major database platforms, including support for Oracle as a metadata platform.

## ***Demonstration Topics***

Document editor and document execution

## ***Milestone Schedule***

Milestone	Description	Target Date
Alpha 3 ZDB	Feature complete for Castor Phase 1. This ZDB will be available for customers for external testing.	6/3/1999
Beta 1 ZDB target	Cleanup tasks complete and software is of sufficient quality for an MSI Way Beta.	7/10/1999

## ***Status***

### ***Recent Progress***

#### **Feature Implementation**

- *Web XML API.* Added support for prompts, report manipulation, generation of graphs for the web, as well as general diagnostics and error handling.
- *Document Object and Execution:* Finished first end-to-end document creation and execution. This is the first stage, which does not include prompts or XML API support.
- *Graph processing for Web.* Integrated the ability to create graphs on the server for display via the Web.
- *Bug Fixing:* We are earnestly fixing bugs for Alpha 3.

## **Stability and Performance**

- *Performance enhancements.* We have made some changes to improve the performance of login authentication. This should also allow us to increase concurrency in general.
- *Memory leak analysis.* For some builds, we have been able to drive our leak tests for basic operations down to zero. We are confident in the results we find from tools like Boundschecker. However, our daily tests using our in-house test programs still reveal memory consumption. We are trying to determine if the issue is with the tests or with the product.
- *Deadlock analysis.* We have removed a number of deadlocks from the system. Nonetheless, as we exercise more code paths, we discover new potential cycles as well. We expect to monitor our progress including an indication of how much code coverage we are achieving with our analysis.
- *All hands stress tests.* The efforts above have been demonstrated in the periodic all-hands stress tests run against the server. Previously, moderate concurrent activity could cause a deadlock on the server after 5-10 minutes of use. Recently, we have been able to survive hour-long stress tests without deadlocks. There are still other bugs and performance issues that are uncovered during these stress tests, but we are seeing progress.
- *Compiler optimization.* We have a dedicated person working on using compiler optimization. We expect many issues to unfold in the course of turning optimization on and Andres Murillo is responsible for driving the effort.

## **Quality Initiatives**

- *Feature testing.* We have given test coverage to all features of the XML API, SQL Cancel, Prioritization, Scheduling, Caching and Cache Administration, Inbox, Clustering.
- *Platform testing.* The 24x7 database rotation program has given us coverage of all databases we plan to support except for Tandem and DB2/390.
- *Metadata certification.* Also, we have tested Oracle 7.3 as a metadata platform and included this in the rotation program. Next steps are DB2, Oracle 8.0, and Oracle 8i.
- *Support for performance analysis.* Ashish Soni continues to support the performance analysis team representing server QE.
- *Alpha site visit to NDC.* Jianhua Wang visited NDC for an Alpha site visit.

## **Performance Analysis**

- *Daily monitoring.* Enterprise Systems Analysis team continues for running daily monitoring tests.
- *Alpha 3 throughput objectives.* We are measuring maximum throughput tests for standard server operations, such as element browsing, object browsing, and report execution (3- and 4-tier). These test results are the ones used for daily monitoring.
- *Alpha 3 concurrency objectives.* We are also measuring user concurrency for the same operations. Bugs in the product that are still under research have hindered these test results.
- *Standard benchmarks.* The Performance Analysis team is defining customer-based scenarios to ensure that our platform can scale to meet customer requirements. Once the concurrency tests are adequately passed, we expect to run a benchmark scenario based on Best Buy's usage profile.

## **Next Steps**

### **Planning**

- *Beta 1.* We will continue the planning process for the Beta 1 development cycle.

### **Feature Development**

- *Web XML API.* Add support for drilling, NT authentication, and anonymous authentication. Add support for Document execution. Enhance Inbox functionality. Enhance support for personalization. Add administrative capability through the XML API.
- *Security:* Integrate multidimensional security into report caching, enhance report subsetting to take advantage of MD security. Finish object access check in server.
- *Report Execution and Caching:* Add intelligent invalidation/update of cache contents.
- *Document Object and Execution:* Add support for prompts and graphs. Optimize for performance.
- *Clustering.* Add cache synchronization and session synchronization to support fail over.

- *Miscellaneous cleanup.* There are a number of miscellaneous feature-related tasks and enhancements that we expect to make during Beta 1.
- *Warehouse Monitor support.* Make changes to statistics tables based on Warehouse Monitor requirements.
- *Broadcaster Aurora support.* Enhance request object and other infrastructure to support Broadcaster integration.
- *Web Deuce support.* Provide translation layer so for Web Deuce product.

#### **Stability and Performance**

- *Performance.* Continue research on compiler optimization and the resolve issues that will inevitably turn up when this occurs.
- *Memory leaks.* Keep basic operations leak-free. Expand coverage to include operations through XML API and more engine features.
- *Stability and deadlocks.* Increase code coverage of deadlock analysis.

#### **Feature Testing**

- Web XML API.
- Document processing.

#### **Quality Initiatives**

- *Close out Alpha 3 milestone.* Ensure that we reach ZDB.
- *Platform testing.* We will tighten up the platform test suites to establish a platform certification process similar to the one we use for the current products. Also, we will continue the platform rotation program.
- *DB2 as metadata platform.* The development team expects to add support for DB2 as a metadata platform in the coming weeks. This will be incorporated into the platform testing plan.
- *Alpha 3 site visits.* The Castor QE team has scheduled site visits in June and Server QE will be involved onsite as well as providing remote support.

#### **Performance Analysis**

- *Alpha 3 objectives.* Measure throughput and concurrency numbers to ensure that we have met our Alpha 3 targets.
- *Benchmark scenarios.* Complete the first iteration of customer scenario benchmarks and take an initial reading.

#### **Issues**

- *Impact of Broadcaster Aurora and Web Deuce.* In order for these development efforts to be successful, the Castor Server must make some changes for appropriate support. These efforts introduce code risk during an important part of our development cycle and also impact our resource allocation for both developers and quality engineers.
- *Project management support.* The server team needs a dedicated project manager to handle resource scheduling for the 12-16 developers and test engineers on the team.
- *Lacking Server QE resources.* The team recently lost one QE and will lose another QE resource in July. In addition, the Kernel engineering team has doubled in size in recent months. Finally, the scope of the server functionality continues to increase as we support Aurora and Deuce. Two Server QE resources should be added.

## Resources

The resource rosters for the Castor Server team are shown below.

Name	Role	Team	Notes
<b>Development Engineers</b>			
Wayne Li	Engineering Manager		
Ramprasad Polana	Software Engineering, team lead	System Debug Team	
Nick Pratt	Software Engineering	System Debug Team	Dead lock detection, performance
Sunil Dixit	Software Engineering, team lead	Stability and Performance Team	
Zheng Wang	Software Test Engineering	Stability and Performance Team	
Lixin Li	Software Test Engineering	Stability and Performance Team	
Juan Muraira	Software Engineering	Stability and Performance Team	
Andres Murillo	Software Engineering	Stability and Performance Team	
Ningning Liu	Software Engineering, team lead	Report Execution, Report Caching	
Yuxiao Xiao	Software Engineering	Report Execution, Graph Engine	
Tina Tian	Software Engineering	Report Execution, Cache admin	
Janaki Goteti	Software Engineering	Web XML API: Execution Flow	
Ping Xu	Software Engineering	Web XML API: Session Manager	
Yuan Ding	Software Engineering	Web XML API: Client component	
Yi Du	Software Engineering	Clustering	
Kevin Wei	Software Engineering	Server security	
Sam Helwig	Software Engineering	Document Object and Document Execution	
Yonghui "Huge" Wang	Software Test Engineering	Build Team	Team build
Longying Zhao	Software Engineering		
<b>Quality Engineers</b>			
Ashish Soni	Quality Engineering	Lead for all backend teams	
Jianhua Wang	Quality Engineering	Lead for Kernel QE	
David Weld	Quality Engineering	Kernel QE	
Dominique Paschoud	Quality Engineering	Kernel QE	
Elsa Polo	Quality Engineering	Kernel QE	
<b>Documentation</b>			
Randy Hechinger	Tech Writer		
<b>Programs</b>			
Scott Cappiello	Program Manager		

## Quality Report

Current issues meeting ZDB criteria for Alpha 3.

TQMS Status	Issues
TBC by QE	13
TBA	19
Assigned	134
Unfixed	0
Ready To Test	45
Postponed	0
<b>TOTAL</b>	<b>232</b>







## Castor Program Status – 7/30/99

Status .....	2
Overall Summary.....	2
Summary – Web .....	2
Summary – COM API & SDK.....	2
Summary – Kernel.....	2
Summary – Engine .....	3
Summary – Interfaces .....	4
Summary – Quality Engineering.....	4
Status Details .....	5
Quality Engineering Detailed Status .....	5
Resources & Roles .....	6
Engine Resources & Roles.....	6
Server Resources & Roles .....	6
COM Resources & Roles.....	7
Interface Resources & Roles.....	8
Web Resources & Roles .....	8
Extended QE Resources & Roles .....	9

## Status

### ***Overall Summary***

July = Moving, planning, company days, and stabilizing Alpha 3. It hasn't been the most productive of months for the technology organization, but all things considered we managed to accomplish a few key goals during a month filled with distractions.

First and foremost, we successfully moved buildings over the span of two weeks – while enduring a week with teams split between the two facilities in the midst of trying to complete our Alpha 3 goal. While the move could have gone a bit smoother, we were able to get settled into our new offices in record time. DSS Labs did a remarkable job of moving all of our servers and getting them operational in the matter of a few days.

During and around the move we focused on rooting out the few remaining bugs in our Alpha 3 build. We have managed to drive our overall bug count down considerably and deliver a solid product. This build gives us a stable platform to build on during our push to Beta 1.

In addition to trying to meet our Alpha 3 goals during July, we have finalized our plans for Beta 1. The plans are designed to finish our remaining feature development while ensuring quality development, and to do all this in an aggressive timeframe. Given our past stabilization efforts we need roughly a month to hunt down and kill our bugs before a major ZDB milestone. Given this, we need to wrap all development up by early to mid September without increasing our bug count significantly. If we can achieve this, we will deliver our Beta 1 build in time for DSS World and be on track to ship the GA software sometime in Q12000.

### ***Summary – Web***

### ***Summary – COM API & SDK***

### ***Summary – Kernel***

The Kernel team efforts are divided into three areas: Execution and Caching, the Web API, and Stability/Performance. Highlights for each subteam during the month of July are listed below.

#### **Execution and Caching**

The Execution and Caching team is primarily completing functionality for document execution and report caching. Recent contributions include caching enhancements to support the Web API inbox work, as well as progress on Cache Invalidation. The latter feature rounds out our cache-matching algorithm so that report caches are not used when the underlying report definition changes.

In addition, this team continued requirements analysis and preliminary design for Broadcaster integration. The scope of this work will not be completed in the Beta 1 timeframe, but the work is independent enough that we will be able to provide a separate development branch for the Broadcaster team to work with. In the past month, the team has worked with the Broadcaster team to complete a requirements document, an interface specification proposal, and a preliminary design document. Next steps are to complete design work and put a development plan in place.

**Web XML API**

The Web API team has drastically revised its development plans in order to reach a September Beta. The team has a significant amount of feature development to complete, and probably represents the critical path to feature completion for the backend. During the month, the team has continued design work for Beta 1 features, including a revised Inbox, Drilling, Searching, and Save as. The team has recently completed API support for NT authentication, anonymous authentication (guest users), the ability to change password, and the ability to display personal folders based on each user. The team is working at a very aggressive pace towards weekly deliverables to support the Web GUI team.

**Stability and Performance**

(Input from Dave Hutz)

The Stability and Performance team is seeing incremental improvements in performance and stability, though we are still short of our Alpha 3 objectives in both areas. The team is finding and fixing lots of good issues, but from a total product level, we are still lacking visibility into how much further we need to go. In happier news, we are making solid progress in memory leaks, meeting our A3 objectives. Members of the team dedicated to tracking memory leaks are moving on to get additional code coverage as planned for Beta 1.

Towards our stability goals, we are nearing the stage where we can run an all-hands-stress through web for object browsing and report execution. It should happen sometime during the week of 8/2. We have uncovered a significant issue in the processing of XSL in our ASP-based products that is a significant problem for Castor Web and probably also Subscriber. It could be as much as a 1 week impact to schedule.

In terms of performance, we should see a 15-35% performance gain in our Alpha 3 branch, as soon as the teams merge into main. There are also some significant improvements (mostly to properties) in Beta scope for COM. The team is cherry-picking some obvious bottlenecks now, but thinks that XML generation is a performance risk (estimating ~30% CPU time in XML generation) and may need to be reworked.

For infrastructure, we are enhancing our suite of automated test tools. Soon, we should be able to setup and run automated Web stress tests, putting more of the test burden on machines.

In addition to supporting the several efforts of the above three teams, the Kernel QE team is responsible for system-wide quality activities. During the past month, Kernel QE began the Customer Project Rotation plan, in which real customer projects are upgraded and tested in regression mode, to broaden our exposure to project-specific quirks. Also, we participated in the Castor-wide mini-regression on A3 candidate builds to flesh out last-minute P1 and P2 issues.

From an overall program perspective, we are facing the following top risks:

- Need visibility into Stability and Performance status. Even if the feature development teams succeed in hitting their aggressive plans, we have a hard time determining whether the product is ready for Beta in terms of performance and stability.
- Need QE support on Web API.

***Summary – Engine***

***Summary – Interfaces***

***Summary – Quality Engineering***

## **Status Details**

### ***Quality Engineering Detailed Status***

## Resources & Roles

### Engine Resources & Roles

Name	Role	Sub Team/ Responsibility
Ben Li	CTA	
Jeff Bedell	Program Management	
Ash Jhaveri	Program Management	
Xinyi Wang	Engineering Team Lead	Analytical Engine
Yuling Ma	Engineering	Analytical Engine
Andrea Torsello	Engineering	Analytical Engine
Xiaonan Han	Engineering	Analytical Engine
Hani Soewandi	Quality	Analytical Engine
Jun Yuan	Engineering Manager	Query Engine, SQL Engine
Xun Feng	Engineering	Query Engine
Yi Luo	Engineering	Query Engine
Parker Zhang	Engineering	Query Engine
Leon Bun	Engineering	SQL Engine
Yinong Chen	Engineering	SQL Engine
Sadanand Sahasrabudhe	Engineering Emeritus (Product Management)	SQL Engine
Lingxiang Chen	Quality	Lead QE
Jun Shun	Quality	SQL Engine
Hank Wang	Quality	Query Engine

### Server Resources & Roles

Name	Role	Sub Team/ Responsibility	Notes
<b>Engineering</b>			
Wayne Li	Engineering Manager		
<b>Stability and performance team</b>			
Ramprasad Polana	Software Engineering	Technical lead	
Nick Pratt	Software Engineering	Development lead	
Zheng Wang	Software Test Engineering		
Lixin Li	Software Test Engineering		
Juan Muraira	Software Engineering		
Yi Du	Software Engineering		
Abhijit Hayatnagarkar	Software Engineering		
<b>Execution and caching team</b>			
Ningning Liu	Software Engineering	Technical lead	
Sam Helwig	Software Engineering	Development lead	
Tina Tian	Software Engineering		
Liqun Jin	Software Engineering		Broadcaster integration
<b>XML API team</b>			
Janaki Goteti	Software Engineering	Technical lead	

Yuan Ding	Software Engineering	Development lead	
Ping Xu	Software Engineering		
Yuxiao Xiao	Software Engineering		
Longying Zhao	Software Engineering		
<b>Build and regression team</b>			
Andres Murillo	Software Engineering	Team lead	
Huge Wang	Software Test Engineering		
<b>Quality Engineering</b>			
Ashish Soni	Quality Engineering	QE lead for all backend teams	
Jianhua Wang	Quality Engineering	QE lead for Kernel team	
Dominique Paschoud	Quality Engineering		
Elsa Polo	Quality Engineering		
Ngone Fall	Quality Engineering		
<b>Documentation</b>			
Randy Hechinger	Tech Writer		
<b>Programs</b>			
Scott Cappiello	Program Manager		
Patrick Vinton	Program Management Engineer	Execution and caching	
David Hutz	Program Manager	Stability and performance	Shared time with Abell products, WH Monitor, Object Manager

**Migration Team**

Name	Role	Sub Team/ Responsibility	Notes
Pat Ori	Programs Engineer	Castor Migration	

**Warehouse Monitor Team**

Name	Role	Sub Team/ Responsibility	Notes
David Hutz	Program Manager	Warehouse Monitor	
Sascha Naujoks	Warehouse Monitor Engineer	Warehouse Monitor	

**COM Resources & Roles**

Name	Role	Sub Team/ Responsibility
Sean McCafferty	Program Manager	Development team project management.
Will Hurwood	Managing Architect	Overall design and architecture for DSS Objects.
Gary Xue	Engineer	Object Management
Zhiying Chen	Engineer	Object Management
Cezary Rascko	Engineer	Object Management
Jing Li	Engineer	Object definitions schema and application.
Dan Preotescu	Engineer	Object definitions and parser development
Ian Falicov	Engineer	Object definitions.
Fabian Camargo	Engineer	Element Browsing, Prompting, Report Resolution
Glenn Boysko	Manager	SDK program management and test engineering
Yansong Wang	Quality Engineer	Object Management, Prompting, Element Browsing
Peter Hefner	Documentation	Developer Guide and API Specification

Jitendra Shirolkar	Software Test Engineer	DSS Web 5.x API Customer Migration, Web API Testing
Lawrence Lun	Software Test Engineer	Drilling, SDK Test Framework/Infrastructure
Lixin Shou	Software Test Engineer	XML Validation (all forms)
Chen Qian	Software Engineer	Application Engineering--Sample Applications
Fernando Gonzalez	Quality Engineer	TQMS Management, Regression Tests, Acceptance Tests

### ***Interface Resources & Roles***

Name	Role	Sub Team/ Responsibility
Fabrice C. Martin	Program Manager	Castor GUI program management
Eduardo Carranza	Engineering Manager	Overall engineering management
Arturo Gay	Engineering Manager	Administration GUI management & engineering
Erika Kuswa	QE Manager	Castor GUI Quality Engineering management
Javier Aldrete	Engineer	Castor Architect Editors design and engineering
Sudhakar Nelamangala	Engineer	Filter Editor & Castor GUI Engineering and design
Jing Ning	Engineer	Administration tools and dialogs design and engineering
Andres Paz	Engineer	Metric Editor & Castor GUI Engineering and design
Sergio Trejo	Engineer	Object Browser and Castor GUI design and engineering
Pankaj Bengani	Quality Engineer	Castor GUI performance quality engineer
Frances Chao	Quality Engineer	Administration quality engineering
Adel Elcheik	Quality Engineer	Metric and Filter functionality quality engineering
Olivia Moncayo	Quality Engineer	Castor Architect quality engineering
Chaitan Kansal	Software Test Engineer	Castor GUI quality engineering
Victor Peña	Engineer	Desktop Viewers
Jorge Garcia	Engineer	Schema Printing Component
Mayra Madrigal	Quality Engineer	Application level editors quality engineering
Hector Aguilera	Quality Engineer	Castor Architect quality engineering
Carlos Madrid	Quality Engineer	Object Browser quality engineering

### ***Web Resources & Roles***

Name	Role
Doug Everhart	Program Manager
Gunther Brenes	Software Architect, GUI Design
Raul Camacho	Engineering Manager
Arturo Oliver	Web GUI Design & implementation, Engineering Manager
Jiefeng Li	Web GUI Design & implementation (XSLs)
Jupiter Munoz	Web GUI Design & implementation (asp/XSL)
Victor Arjona	Software Engineer
Andrew Smith	QE Lead
Alda Cheng	Quality Engineer
Jonathan Jiang	QE

### ***Extended Web Team***

Name	Role
Wayne Li	Web Server, Server implementation
Janaki	Web Execution flow
Sam Helwig, NingNing	Document Object
Will Hurwood, Jin Li, Zhiying Chen, Fabian Camargo	COM designs and implementation for Web
Yuxiao Xiao	Graph object



Glenn Boysko, Jitendra Shirolkar	Review of the Web Script Library and additional support (as the SDK Team has its own website based on the Web API and the Web Team's script library).
Yuan Ding	Web Server side implementation

### ***Extended QE Resources & Roles***

#### **QE Integration Team**

Name	Role
Olivier Marchal	QE Lead
Mala Viswanath	Installation, configuration wizard, and diagnostics (for Beta 1)
José Rosas	End-to-end story

#### **QE Cross Team**

Name	Role
Ana López	QE Release Manager
Dan Kerzner	Alpha and Beta Programs Lead
Cuong Bui	VMALL Engineer



## Castor Program Status – 8/27/99

Status.....	2
Overall Summary.....	2
Summary – Web .....	2
Summary – COM API & SDK.....	2
Summary – Kernel.....	2
Summary – Engine .....	3
Summary – Interfaces .....	4
Summary – Quality Engineering.....	4
Status Details .....	5
Quality Engineering Detailed Status .....	5
Resources & Roles .....	6
Engine Resources & Roles.....	6
Server Resources & Roles .....	6
COM Resources & Roles.....	7
Interface Resources & Roles.....	7
Web Resources & Roles .....	7
Extended QE Resources & Roles .....	7

**Status*****Overall Summary******Summary – Web******Summary – COM API & SDK******Summary – Kernel***

The Kernel team efforts are divided into three areas: Execution and Caching, the Web API, and Stability/Performance. Highlights for each subteam during the month of August are listed below.

**Execution and Caching**

The execution and caching team has had a steady and consistent development process the past month. The team added Inbox, graph, drilling, pivoting, and sorting support to documents, which ultimately completes backend document support for Web. The team also refined the report cache matching algorithm to include prompt comparison for reports with filter or template prompts.

In order to complete the Beta 1 build, the team will continue development on cache synchronization in a clustered environment and will enhance the client connection mechanisms. After this is complete, no new feature development is planned. Proactively, the team will then focus its efforts on performance; reactively, the team will respond to QE scalability tests and wrap up small details on features to complete story-lines.

Working in parallel to the rest of the team on a separate timeline, Liquan Jin has worked with the Broadcaster team to design a Server interface, and is currently half way done with its development. This work has been put on a separate timeline because it is not critical to the beta releases of the product since Broadcaster Aurora development will be still in infant stages at this time. However, this new interface is expected to be complete in two weeks, and QE is already poised to begin testing then.

**Web XML API**

The month of August was a very aggressive development period for the Web API team. The team executed on an ambitious five-week plan for delivering weekly functionality for the Web GUI team. Execution success was due to well-thought-out designs, frequent communication in daily meetings, and the hard work of engineers and QEs working long hours and weekends.

Key features now available through the API include a completely revised Inbox, which is the basis for a significant number of web features; drilling support for reports; support for drilling, prompts, and graphs with documents; searching; "save as" support for reports; and administration of web-specific settings.

Looking ahead, the Web API team plans to focus on stability and performance of the Web API. In addition to addressing any issues uncovered by the Web GUI team and by QE testing, the team will engage in code reviews and stress testing to improve the stability of the product.

### **Stability and Performance**

The Stability and Performance team continues to find and fix lots of good issues, but from a total product perspective, there is a great deal of work to do. The product overall has fallen short of our Alpha 3 objectives for Performance and Stability. We had hoped at this point to be able to survive an all-hands stress test in either 3-tier or 4-tier. We have succeeded with automated tests using simulated clients, and the server has survived both 3-tier and 4-tier manual stress tests, but we continue to identify new issues with each test that prevent us from really stressing the server in the all-hands scenario. In terms of performance, the A3 build is close to our performance targets for low user concurrency, but well off the mark for high concurrency.

In addition to performance and stress objectives, the team is also responsible for memory leak analysis and for code/design reviews of key modules critical to overall server stability. During the month of August, the team expanded the scope of automated memory leak tests to include additional operations: server admin commands, inbox retrieval, and document execution. The team also optimized the service manager module to improve performance for dealing with requests to the server. Finally, the team implemented major changes to the database connection modules to improve robustness and error-handling capabilities.

As we move towards the Beta release, the team has produced itemized lists of changes and optimizations for both stability and performance. In addition to proceeding through these lists, the team will continue to conduct regular all-hands stress tests to generate new issues. For memory leak analysis, the team is trying to shift its role from fixing memory leaks to identifying leaks and providing infrastructure so that the actual resolution may be distributed to engineering teams. The Enterprise Analysis team will continue to run daily tests to monitor performance on a build-by-build basis and also monitor progress in terms of the I-Benchmark goal for Beta 1.

The Server QE team supports the development efforts of each of the above teams. QE has been testing features for the Exec/Caching team, such as the use of security filters with caching. QE also played an instrumental role as the gatekeeper of the XML API team's weekly deliverables, validating all features mentioned above. In addition to supporting these efforts, the Server QE team is responsible for system-wide quality activities. During the past month, Server QE ran regression tests to help us close out feature issues for Alpha 3, while also updating the suite of regression tests. We also continued the customer project rotation program, looking for issues with the infamous Western Digital project.

From an overall program perspective, we are facing the following top risks:

- Need better visibility into performance and stability. We need to get better at measuring where we are and how close we are to where we would like to be. We know we are making progress at the micro level where a lot of fixes are made, but the bigger picture is not as clear.
- Performance analysis to date has neglected response time objectives, particularly for three-tier. While we may make progress with throughput numbers, the perceived response time of the product continues to be an issue.
- Direct QE resource is needed for stability work. This role is being played in part by Ashish Soni and members of the EA team. However, the engineers need dedicated QE resources who can work more closely with engineers.

### **Summary – Engine**

***Summary – Interfaces***

***Summary – Quality Engineering***

**Status Details**

***Quality Engineering Detailed Status***

## Resources & Roles

### Engine Resources & Roles

### Server Resources & Roles

Name	Role	Sub Team/ Responsibility	Notes
<b>Engineering</b>			
Wayne Li	Engineering Manager		
<b>Stability and performance team</b>			
Ramprasad Polana	Software Engineering	Technical lead	
Nick Pratt	Software Engineering	Development lead	
Zheng Wang	Software Test Engineering		
Lixin Li	Software Test Engineering		
Juan Muraira	Software Engineering		
Abhijit Hayatnagarkar	Software Engineering		
<b>Build and regression team</b>			
Andres Murillo	Software Engineering	Team lead	
Huge Wang	Software Test Engineering		
<b>Execution and caching team</b>			
Ningning Liu	Software Engineering	Technical lead	
Sam Helwig	Software Engineering	Development lead	
Tina Tian	Software Engineering		
Liqun Jin	Software Engineering		Broadcaster integration
<b>XML API team</b>			
Janaki Goteti	Software Engineering	Technical lead	
Yuan Ding	Software Engineering	Development lead	
Ping Xu	Software Engineering		
Yuxiao Xiao	Software Engineering		
Longying Zhao	Software Engineering		
Yi Du	Software Engineering		
<b>Quality Engineering</b>			
Ashish Soni	Quality Engineering	QE lead for all backend teams	
Jianhua Wang	Quality Engineering	QE lead for Kernel team	
Dominique Paschoud	Quality Engineering		
Elsa Polo	Quality Engineering		
Ngone Fall	Quality Engineering		
<b>Documentation</b>			
Randy Hechinger	Tech Writer		
<b>Programs</b>			
Scott Cappiello	Program Manager		
Patrick Vinton	Program Management Engineer	Execution and caching	
David Hutz	Program Manager	Performance analysis	Shared time with Abell products, WH Monitor, Object Manager

### Migration Team

Name	Role	Sub Team/ Responsibility	Notes
Pat Orie	Programs Engineer	Castor Migration	

***Warehouse Monitor Team***

Name	Role	Sub Team/ Responsibility	Notes
David Hutz	Program Manager	Warehouse Monitor	Shares time with Abell products, Performance Analysis, Object Manager
Sascha Naujoks	Warehouse Monitor Engineer	Warehouse Monitor	

***COM Resources & Roles******Interface Resources & Roles******Web Resources & Roles******Extended QE Resources & Roles***





## Castor Program Status – 9/24/99

Status .....	2
Overall Summary .....	2
Summary – Web .....	2
Summary – COM API & SDK .....	2
Summary – Kernel .....	2
Summary – Engine .....	3
Summary – Interfaces .....	3
Summary – Quality Engineering .....	3
Status Details .....	4
Quality Engineering Detailed Status .....	4
Resources & Roles .....	5
Engine Resources & Roles .....	5
Server Resources & Roles .....	5
COM Resources & Roles .....	6
Interface Resources & Roles .....	6
Web Resources & Roles .....	6
Extended QE Resources & Roles .....	6

## **Status**

### ***Overall Summary***

#### ***Summary – Web***

#### ***Summary – COM API & SDK***

#### ***Summary – Kernel***

The Kernel team is divided into three subteams: Execution Flow team, the XML API team, and the Stability/Performance team. Highlights for each subteam during the month of September are listed below.

##### **Execution and Caching**

The Execution and Caching team has continued their steady development process the past month. All Beta 1 features are complete, so the team can now concentrate keeping the bug count low and cleaning up its code.

The team is now gearing up for the Beta 2 cycle. Most planned tasks include code cleanup for stability and performance, such as dividing the Report Server and Report Instance (for stability and ease of maintenance) and tuning user authentication mechanisms (for performance).

The team also does plan to do some feature development in Beta 2. The marquee Beta 2 feature is Broadcaster Aurora support. Although this is a fairly significant feature that typically would not be part of beta development, this work is reasonably independent of code used by the Desktop and Web. The remaining Beta 2 feature development is minor but necessary to complete end-to-end stories and preserve backward compatibility with the Abell product suite.

##### **Web XML API**

The first half of September was focused on getting the features developed during August into the main build. After an arduous merge process, the main build now supports the Beta 1 web feature set in its entirety, free of memory leaks, and passing regression testing. In the second part of the month, the XML API team turned its attention to stability and performance and fixing TQMS issues. Now that the main build supports all web features, we are able to run 4-tier stress tests and these tests have uncovered several useful issues for the team. In addition, we are beginning to reduce the overall issue count, although some QE feature testing is still outstanding.

Looking ahead, the Web API team plans to concentrate on TQMS issues and on stability and performance of the Web API. The team is also planning for the Beta 2 development cycle. Once again, this team probably has the heaviest load of planned development for the cycle. Although the planned work is not nearly as extensive as the Beta 1 cycle, there are "loose end" features to support the Web GUI and two minor enhancements to support the SDK team's analysis of requirements of existing Web API users.

## **Stability and Performance**

Last month, one of the top issues was gaining visibility into the progress of the performance and stability team, since a lot of their work is interrupt-driven. We have developed a set of management controls that the team reviews with Steve Trundle and the project management team on a weekly basis. These controls are aligned with the key initiatives of the team:

- Enforcing the adoption of coding standards throughout the backend teams.
- Monitoring and resolving potential deadlock situations in our software.
- Running semi-weekly all-hands stress tests and resolving resulting issues.
- Monitoring and enforcing the resolution of memory leaks and memory usage in the server.

Now that the 4-tier features are available in the main build and we have added Sumeet Bhalla on the QE team, we are able to add the following additional initiatives:

- Running automated stress tests on a regular basis.
- Running stability tests that target boundary cases in the server.

For the Beta 2 cycle, the marquee feature contribution of the stability and performance team will be revised database connection code, which has been enhanced for stability and robustness for a critical part of the server's responsibility.

## **Quality Engineering**

The Server QE team supports the development efforts of each of the above teams. QE expects to finish Beta 1 feature testing by the end of September, including cache synchronization for clustering and database certification. Most of the next month will be spent reaching the quality objectives for Beta 1.

## **Risks/Issues**

From an overall program perspective, we are facing the following top risks:

- Progress on stability work needs to accelerate. We are making progress and have better controls to monitor the progress, but it is clear that we need to move faster. A lot of the stability issues that we turn up in our various stress tests do not make their way to TQMS, so metrics based on TQMS issue counts can be misleading in judging stability of the product.
- Performance analysis to date has neglected response time objectives, particularly for three-tier. While we may make progress with throughput numbers, the perceived response time of the product continues to be an issue.
- Need to keep tight control on scope. Although each subteam knows of some additional development work to be done, we need to carefully limit the amount of code changes we allow at this stage of development.

## ***Summary – Engine***

## ***Summary – Interfaces***

## ***Summary – Quality Engineering***

## Status Details

### Quality Engineering Detailed Status

### Stability and Performance Management Scorecard

#### Deadlocks

<b>Deadlock Metric</b>	<b>Current</b>	<b>Beta 1 Goal</b>
Known cycles	9	0
Code exercised	Automated Stress Test	Full Regression

#### Stability Tests

<b>Metric</b>	<b>Current</b>	<b>Beta 1 Goal</b>
Percent of cases tested	0%	N/A
Percent of cases passed	0%	N/A

- New QE joined Server team 9/21 to begin executing these tests.

#### Automated Stress Tests (see [Simulated stress scorecard](#) below)

- Most successful test to date:
  - 500 users, ~6000 jobs, 25 minutes
  - 4-tier: report execution

#### All-hands Stress Tests (see [All-hands stress scorecard](#) below)

- Last all-hands stress test:
  - 101 users, 700 jobs
  - 4-tier: report execution, inbox operations, prompt execution, report caching

#### Memory Leaks and Usage (see [Memory Leak scorecard](#) below)

<b>Memory Metric</b>	<b>Leak</b>	<b>Usage</b>	<b>Beta 1 Goal</b>
Scenarios in automated tests	94%	56%	100%
Scenarios assigned owners	97%	97%	100%
Tested scenarios that are clean	87%	66%	100%
Total scenarios that are clean	81%	38%	100%

#### Performance

<b>Performance Benchmark</b>	<b>Current Status</b>	<b>Beta 1 Goal</b>	<b>GA Goal</b>	<b>Abell</b>
I-Benchmark-Beta Throughput	Test fails	300 rpm	N/A	

## Resources & Roles

### Engine Resources & Roles

### Server Resources & Roles

Name	Role	Sub Team/ Responsibility	Notes
<b>Engineering</b>			
Wayne Li	Engineering Manager		
<b>Stability and performance team</b>			
Ramprasad Polana	Software Engineering	Technical lead	
Nick Pratt	Software Engineering	Development lead	
Zheng Wang	Software Test Engineering		
Lixin Li	Software Test Engineering		
Juan Muraira	Software Engineering		
Abhijit Hayatnagarkar	Software Engineering		
<b>Build and regression team</b>			
Andres Murillo	Software Engineering	Team lead	
Huge Wang	Software Test Engineering		
<b>Execution Flow team</b>			
Ningning Liu	Software Engineering	Technical lead	
Sam Helwig	Software Engineering	Development lead	
Tina Tian	Software Engineering		
Liquan Jin	Software Engineering		Broadcaster integration
<b>XML API team</b>			
Janaki Goteti	Software Engineering	Technical lead	
Yuan Ding	Software Engineering	Development lead	
Ping Xu	Software Engineering		
Yuxiao Xiao	Software Engineering		
Longying Zhao	Software Engineering		
Yi Du	Software Engineering		
<b>Quality Engineering</b>			
Ashish Soni	Quality Engineering	QE lead for all backend teams	
Jianhua Wang	Quality Engineering	QE lead for Kernel team	
Dominique Paschoud	Quality Engineering		XML API
Elsa Polo	Quality Engineering		Execution Flow
Ngone Fall	Quality Engineering		Execution Flow
Sumeet Bhalla	Quality Engineering		Stability and Performance
<b>Documentation</b>			
Randy Hechinger	Tech Writer		
<b>Programs</b>			
Scott Cappiello	Program Manager		
Patrick Vinton	Program Management Engineer	Execution and caching	
David Hutz	Program Manager	Performance analysis	Shared time with Abell products, WH Monitor, Object Manager

### Migration Team

Name	Role	Sub Team/ Responsibility	Notes
Pat Orie	Programs Engineer	Castor Migration	

***Warehouse Monitor Team***

Name	Role	Sub Team/ Responsibility	Notes
David Hutz	Program Manager	Warehouse Monitor	Shares time with Abell products, Performance Analysis, Object Manager
Sascha Naujoks	Warehouse Monitor Engineer	Warehouse Monitor	

***COM Resources & Roles******Interface Resources & Roles******Web Resources & Roles******Extended QE Resources & Roles***



## Castor Program Status – 11/22/99

Status .....	2
Overall Summary .....	2
Summary – Web .....	2
Summary – COM API & SDK .....	2
Summary – Kernel .....	2
Summary – Engine .....	3
Summary – Interfaces .....	3
Summary – Quality Engineering .....	3
Status Details .....	4
Quality Engineering Detailed Status .....	4
Resources & Roles .....	4
Engine Resources & Roles .....	4
Server Resources & Roles .....	4
COM Resources & Roles .....	5
Interface Resources & Roles .....	5
Web Resources & Roles .....	5
Extended QE Resources & Roles .....	5

## Status

### ***Overall Summary***

#### ***Summary – Web***

#### ***Summary – COM API & SDK***

#### ***Summary – Kernel***

The Kernel team is divided into three subteams: Execution Flow team, the XML API team, and the Stability/Performance team. Ordinarily, these subteams operate fairly independently. Like all backend teams during the past month, all three have been focused on reaching the Beta 1 Performance and Stability objectives.

#### **Stability and Performance**

The charter of the Stability and Performance team has been to serve as a clearinghouse for all backend teams with regard to the multiple performance and stability objectives. These objectives include targets for

- eliminating potential deadlocks from the system,
- reducing memory leaks and memory usage for key operations,
- stressing the server with automated tools,
- stressing the server with manual usage, and
- achieving acceptable scores on the I-Benchmark.

As of 11/19, all objectives are still in progress. Deadlocks and memory usage seem to have reasonably predictable paths to completion. We still have difficulty predicting when we can reach the stress testing objectives.

#### **Top Issues**

From an overall program perspective, we are facing the following top issues in addition to those already discussed in the Quality Engineering section:

- *Ability to execute towards cross-team objectives.* During the course of the past cycle, we have had difficulty reaching objectives that require work across all backend teams. There are a number of contributing factors: we seem to be lacking aggressive owners who can drive cross-team issues, owners lack perceived authority to rally the resources they need, and the team as a whole tends to get distracted when working towards multiple objectives.
- *Level of confidence in our ability to hit stability and performance goals in general.* Since it has taken more effort than anticipated to hit the Beta 1 objectives, we need to adjust our plans for stability and performance work in the next development cycle. We need to devote more of our time and resources towards such objectives and make sure that we can effectively drive the work across teams.



- *Impact of Beta 1 slippage on Beta 2 plans.* Given the issue mentioned above, as well as the time lost trying to close the Beta 1 objectives, we have had to revisit our strategy for Beta 2. While the majority of the remaining for Beta 2 falls in the category of bug fixing and stabilization, there are a set of tasks to complete unfinished features. Instead of completing these features, we expect to remove them outright. Features at risk include support for clustering with Inbox synchronization, server statistics on DB2 and Oracle, 3-tier project creation, support for Broadcaster Aurora, and Web API enhancements for customers who are migrating.

### ***Summary – Engine***

### ***Summary – Interfaces***

### ***Summary – Quality Engineering***

## Status Details

### Quality Engineering Detailed Status

### Stability and Performance Management Scorecard

## Resources & Roles

### Engine Resources & Roles

### Server Resources & Roles

Name	Role	Sub Team/ Responsibility	Notes
<b>Engineering</b>			
Wayne Li	Engineering Manager		
<b>Stability and performance team</b>			
Ramprasad Polana	Software Engineering	Technical lead	
Zheng Wang	Software Test Engineering		
Lixin Li	Software Test Engineering		
Abhijit Hayatnagarkar	Software Engineering		In Bootcamp
<b>Execution Flow team</b>			
Ningning Liu	Software Engineering	Technical lead	
Sam Helwig	Software Engineering	Development lead	
Tina Tian	Software Engineering		
Liqun Jin	Software Engineering		Broadcaster integration
<b>XML API team</b>			
Janaki Goteti	Software Engineering	Technical lead	
Yuan Ding	Software Engineering	Development lead	
Ping Xu	Software Engineering		
Yuxiao Xiao	Software Engineering		
Longying Zhao	Software Engineering		
Yi Du	Software Engineering		
<b>Quality Engineering</b>			
Ashish Soni	Quality Engineering	QE lead for all backend teams	
Jianhua Wang	Quality Engineering	QE lead for Kernel team	
Dominique Paschoud	Quality Engineering		XML API
Elsa Polo	Quality Engineering		Execution Flow
Ngone Fall	Quality Engineering		Execution Flow
Sumeet Bhalla	Quality Engineering		Stability and Performance
Hengky Suryadi	Quality Engineering		Acceptance
<b>Documentation</b>			
Randy Hechinger	Tech Writer		
<b>Programs</b>			
Scott Cappiello	Program Manager		
Patrick Vinton	Program Management Engineer	Execution and caching	

***System Component Team***

Name	Role	Sub Team/ Responsibility	Notes
Doug Meyer	Engineering Manager		
Nick Pratt	Software Engineering		
Juan Muraira	Software Engineering		
Andres Murillo	Software Engineering		
Javier Leija	Software Engineering		Testing architecture

***Migration Team***

Name	Role	Sub Team/ Responsibility	Notes
Pat Orié	Programs Engineer	Castor Migration	

***Warehouse Monitor Team***

Name	Role	Sub Team/ Responsibility	Notes
Sascha Naujoks	Warehouse Monitor Engineer	Warehouse Monitor	

***COM Resources & Roles******Interface Resources & Roles******Web Resources & Roles******Extended QE Resources & Roles***



## Castor Program Status – 11/22/99

Status.....	2
Overall Summary .....	2
Summary – Engine.....	2
Summary – Kernel .....	2
Summary – COM API.....	3
Summary – Desktop.....	3
Summary – Web .....	4
Summary – Quality Engineering.....	4
Summary – Migration .....	5
Quality Report.....	6
Castor Mercury Assessment .....	9
Castor Beta Programs: Site Overview.....	12
Castor Beta Programs: Site Visit Schedule.....	13
Castor Beta Programs: Site Status .....	14
Castor Resources & Roles.....	16
Engine Resources & Roles .....	16
Server Resources & Roles.....	17
COM Resources & Roles .....	18
Desktop Resources & Roles .....	18
Web Resources & Roles .....	19
Extended QE Resources & Roles .....	19

## **Status**

### ***Overall Summary***

The past two months have been exciting and frustrating at the same time. The team is still driving towards its Beta 1 milestone, a mark that we had hoped to achieve by MicroStrategy World. While the team was in fairly good shape with feature development at the time of the last review, we have not been able to achieve our performance and stability goals required to call it Beta 1. This reality has been extremely frustrating for everyone involved and still plagues us. The team continues to test and attempt to isolate the root of the instability problems though we are still uncertain as to when we will achieve the Beta 1 milestone as we had initially defined it.

The bright spots over the last couple months relate to the Castor presentations we have given. During the week of MicroStrategy world we ran two successful product demonstrations that went off without a hitch – one during the company meetings and another during Futures Day. The Futures Day presentation was received with many rounds of applause indicating customers' approval of the product's new feature set. Additionally, we showed the product at the Spanish User Conference in La Toja and at the European Company days in London. As with the previous presentations, Castor showed well and generated some positive excitement.

Given that the product is relatively stable – stable enough to conduct customer site visits, but not stable enough to meet our B1 criteria – we have begun our B1 customer testing. This has allowed us to continue pushing our testing cycle forward in spite of still being officially in Alpha. At this point we have visited six different sites including our dear friends at La Caixa. So far testing has gone well (more detail to follow).

In short, we are in crunch mode with Castor. The interface teams are working on Beta 2 development and are only somewhat impeded by the delay in the back end. The server teams have been directed to focus solely on stability and testing, and will hopefully get the product in Beta shape in the near future.

### ***Summary – Engine***

The engine team has completed almost all of the feature work for Castor. The few remaining pieces of functionality are more bug fixing than feature development, with a primary area being SQL optimization. As part of the beta testing, we are gathering test data about the performance and syntax differences between Abell and Castor SQL. We expect to be added SQL and general processing optimizations into engine during entire course of beta.

In addition, the team is helping out with stability testing and performing code reviews. Many of the issues with SQL and engine optimization relate to assumptions made in the initial design of the code. Consequently, these require review and some design work before making any changes. No changes will be made until the server achieves Beta 2 though design work on these optimization “bugs” is ongoing.

Along with the beta testing, the team continues to perform customer project testing and feature combination testing. As we begin to stress the engine we are finding more and more issues which is to be expected. As a result, we will continue to expand our test coverage as quickly as possible.

### ***Summary – Kernel***

The Kernel team is divided into three subteams: Execution Flow team, the XML API team, and the Stability/Performance team. Ordinarily, these subteams operate fairly independently. Like all backend teams during the past month, all three have been focused on reaching the Beta 1 Performance and Stability objectives.

## ***Stability and Performance***

The charter of the Stability and Performance team has been to serve as a clearinghouse for all backend teams with regard to the multiple performance and stability objectives. These objectives include targets for

- eliminating potential deadlocks from the system,
- reducing memory leaks and memory usage for key operations,
- stressing the server with automated tools,
- stressing the server with manual usage, and
- achieving acceptable scores on the I-Benchmark.

As of 11/19, all objectives are still in progress. Deadlocks and memory usage seem to have reasonably predictable paths to completion. We still have difficulty predicting when we can reach the stress testing objectives.

## ***Top Issues***

From an overall program perspective, we are facing the following top issues in addition to those already discussed in the Quality Engineering section:

- ***Ability to execute towards cross-team objectives.*** During the course of the past cycle, we have had difficulty reaching objectives that require work across all backend teams. There are a number of contributing factors: we seem to be lacking aggressive owners who can drive cross-team issues, owners lack perceived authority to rally the resources they need, and the team as a whole tends to get distracted when working towards multiple objectives.
- ***Level of confidence in our ability to hit stability and performance goals in general.*** Since it has taken more effort than anticipated to hit the Beta 1 objectives, we need to adjust our plans for stability and performance work in the next development cycle. We need to devote more of our time and resources towards such objectives and make sure that we can effectively drive the work across teams.
- ***Impact of Beta 1 slippage on Beta 2 plans.*** Given the issue mentioned above, as well as the time lost trying to close the Beta 1 objectives, we have had to revisit our strategy for Beta 2. While the majority of the remaining for Beta 2 falls in the category of bug fixing and stabilization, there is a set of tasks to complete unfinished features. Instead of completing these features, we expect to remove them outright. Features at risk include support for clustering with Inbox synchronization, server statistics on DB2 and Oracle, 3-tier project creation, support for Broadcaster Aurora, and Web API enhancements for customers who are migrating.

## ***Summary – COM API***

The COM Team spent the month of November reaching and maintaining ZDB as well as participating in the memory usage and deadlock elimination efforts run by Kernel. We have been successful in maintaining our Beta 1 ZDB status and have eliminated all deadlocks assigned to the COM team. We are currently running daily stress tests to assist in find Access Violations and tracking down memory usage scenarios. Our plan for Beta 2 is set. It contains minor enhancements, bug fixes and API documentation reviews. We are just beginning the planning stage for Castor 2. Our goal for the next two months is to begin beta 2 development, lock down Castor 2 plans so that we can begin the design process and continue bug fixing and eliminating poor memory usage scenarios.

## ***Summary – Desktop***

The Desktop team closed successfully the Beta 1 milestone and development cycle a couple of weeks ago and has now completely moved to Beta 2 development. A good part of our senior engineers and quality engineers will be travelling at least for some days during the Beta 1 period, testing and presenting their work and obtaining direct feedback from our customers on the different targeted sites.

On the Beta 2 front, as we did for Beta 1 our initial focus will be purely on stabilization and robustness work. The top team-wide priority will be to get to a ZDB (zero defect build) status as soon as possible. The team will also focus during at least some part of the Beta 2 time in performing extensive code reviews of some of the "high exposure" components. Once we reach a level of comfort in terms of product stability and performance and if time permits, we will attack a few of the enhancements that have come up during the different feature reviews, usability labs and customer visits.

### ***Summary – Web***

The Web GUI team successfully completed the Beta 1 milestone in compliance with MSI Way guidelines. The team is now moving forward with Beta 2 plans, while providing support for the backend teams as needed. Coming in Beta 2 is an update to the look and feel theme of the interface, based on customer input. The team will also complete loose ends on remaining functionality. Finally, the team will focus on code cleanup and tackle performance optimizations.

### ***Summary – Quality Engineering***

All planned QE tasks for accomplishing an MSI Way Beta 1 release by DSSWorld were carried out. However, we were unable to achieve this goal due to stability issues.

An 'assessment' document was developed as a checklist for B1 readiness, in terms of site visits and compliance with the MSI Way. The latest copy is available at the end of the QE section.

In the period between 7/31/99 and today, the Castor team has resolved these issues (defects, enhancements, feature development):

Severity	Number of Issues
1	131
2	655
3	2065
<b>Total</b>	<b>2851</b>

This past month, the backend QE teams (Engine, COM, Kernel) got involved in the task forces set up to address stability issues (stress tests and memory usage tests).

Acceptance rotation has proven to be very good training for new Castor QE's. We'll continue this practice, in coordination with the QE Cross-Team led by Olivier and the Build team led by Jim Bennett.

QE worked with APS and Beta Programs to define the procedure to process issues coming from the field.

The site visits coordinated by Beta Programs kicked off on 11/8/99. Issues have started to trickle in, and both QE's and Engineers have been involved in the visits.

The sign off documents for the Castor product suite have been drafted and we've begun the process of reviewing each of them, with the understanding that though a team may be ready for Beta 1, we cannot declare victory until we achieve the Stability goals.

In the next week, QE will finalize the plans for B2. These will account for the recent addition of Japanese to the roster of languages to be certified by GA. There will be a lag between the execution of the plans from the backend teams, since they will continue to work on wrapping up B1 stability goals. All teams will continue to support our Beta effort, interacting with APS, Beta Consultants, and the clients.

***Summary – Migration***

After months of preparation, we are beginning to see the fruits of our efforts as the Castor Beta Preview product was released to a handful of our customers for Beta testing on 11/8/99. Most of the migration issues that we encountered in the first 2 weeks of Beta 1 we anticipated. These issues need to be addressed further as we go forward.

- Lack of Excel Workbook functionality on Web
- ActiveX flexibility (drilling and outline mode) on Web
- VLDB properties not upgraded to Castor
- Castor repository sizing unknowns
- SQL inconsistencies

Over the next few months we expect to encounter other migration issues that once addressed will help us narrow the gap between Abell and Castor. Concentration on enhancing the Castor Upgrade Manual, Technical Support expertise, and the Castor KBase will be critical as we continue the Beta 1 testing.

As we approach Beta 2, migration will become more externally focused. We need to be prepared from both a product and education standpoint so that all customers and MSI personnel who evaluate the product do so successfully. To assist in this effort, we will need to work closely with Education to ensure that the Castor Migration course is completed and ready for delivery and that all materials accompanying the migration are updated and accurate.



## **Quality Report**

### **Recent progress**

#### **Release Management**

##### **Highlights:**

- Beta Program officially started on 11/8/99. QE's and Engineers have been visiting client sites during the month.
- Ran an Installation BugFest to certify the B1 install in international environments and US English.
- Coordinated creation of sign off documents for Beta 1.
- New assignment: University Week courses for PUC (Product Use Competition)
- Vmall – Working on adding a customized Web 'store front'. Added document objects.
- StockMarket – integration of ASP pages for document objects, more reports.
- Affiliate Reporting Project for Strategy.com – now in production.
- Corporate\_Dev – use of ASPs, Web API, and XML
- TQMS on Castor – reports for sign off documents, use of caching.

##### **Lowlights:**

- Missing Beta 1 stability goals.
- Possible scope creep due to addition of Japanese.

#### **Server**

##### **Highlights:**

- Have QE 100% devoted to server stability.
- Server lasts longer and better under stress condition.
- QE begin to setup automated test infrastructure, every QE need to automate his(her) test to reduce the man-hour spend on repetitive regression and acceptance test. The entire test program developed will confirm to a same SDK standard.

##### **Lowlights:**

- Not Beta yet.

#### **COM**

##### **Highlights:**

- TQMS issues were under control in COM QE side (It was a big concern for last month).
- New members speeded up in handling QE work.
- New Test development environment in ClearCase has been set and ready for developers to write unit tests.

##### **Lowlights:**

- No new tests are developed yet
- Distracted from Memory leak tests.
- The document for writing COM API tests got delayed.

#### **Engine**

##### **Highlights:**

- New member Gong Rui joined Engine QE team
- Engine feature combination testing has started. More than 30 issues have been found. We expect this test will last 4 months and greatly improve Engine quality.

- Two recent fixes in Engine code have greatly reduced the Castor number of intermediate tables to Abell level in Metro project;
- The total number of Engine Acceptance reports is now 1,500
- The Engine team has been working on memory leak. The goal is to certify all 1,500 Engine acceptance reports.

**Lowlights:**

- Engine performance is an important issue now;
- We have conducted only less than 10% of the feature combination test, but already found lots of issue, we might have problem fixing all these issues

**Interfaces****Highlights:**

- The GUI is caught up as far as TQMS issues go which means more stability.
- QE is automating more tests in order to cover more in less time.
- Initial feedback from beta has been productive.

**Lowlights:**

- No new lowlights aside from the fact that we still have a lot to do.

**Web****Highlights:**

- We are considered MSI way compliant Beta1 for Castor Web
- Got a lot of great feedback from Customers how to drive the new interface direction
- Estee Site visit was very insightful for migration issues

**Lowlights:**

- Performance like a pig
- ActiveX faithful customers are not ecstatic about all-html in Castor Web

**Cross team****Highlights:**

- All the goals we were given for Beta1 have been achieved except for the cleaning of the COM diagnostics (lower priority)
- The Kernel, COM and GUI branches now run a fully automated GUI script as part of their acceptance
- We have started getting several GUI team members to use our new GUI automation tool
- We have started rethinking our acceptance processes in parallel with the build processes

**Lowlights:**

- We are still far from a model in which the build when ready would kick-off different set of tests (memory leaks, stress tests, XML and COM API regression + GUI regression) and report the results in a single file. Thus we require human intervention, and acceptances are felt as resource-consuming...

**Next steps**

The focus for QE in November and December will be wrapping up B1 (stability) and start work on Beta 2. The summarized goals are:

1. Stability (stress tests, memory leaks, performance)
2. Feature testing for Beta 2 features (postponed Beta 1 scope + issues from client site testing)
3. Regression testing for most features (major focus on automation).
4. Platform testing (additional DBs, languages, OS).

5. Internal Beta sites – MSI DSS applications + projects from customers.
6. Regular QE tasks (TQMS, support site visits, support APS and Beta consultants, etc.)

## Issues

### Major risks for Beta 1:

- Stability

### Major risks for Beta 2:

- There are several Engineering plans for Beta 2 that have not been defined: XML API, Stability, Object Manager, SDK, and Diagnostics. QE needs this information to prepare for Beta 2 adequately.
- Japanese – this hasn't been fully accounted for in the Engineering plans yet.
- Stability – we need all teams to continue to allocate resources to this effort throughout the milestone, not just at the end.
- Supporting Beta sites adequately. We've define procedures and have been working with Tech Services to have as many of the client projects in-house as possible, to alleviate this risk. However, when we get to Beta 2, it will become an open enrollment program, with up to 70 clients signed on.
- Licensing of IE 5.0 DLLs components remains an open issue. For Beta 1, we will require IE 5.0 to be installed on clients and server machines.
- Number of issues open for Mercury GA: We have **356** issues open for engineering, **1117** issues open for Program Management, and **343** issues open for QE. This is before QE even begins to test B2 features, or the Beta program is in full swing.

- QE in action

Status	1	2	3	Total
Ready to Test	5	27	180	212
TBC by QE	1	4	126	131
<b>Total</b>	<b>6</b>	<b>31</b>	<b>306</b>	<b>343</b>

- Engineering in action

Status	1	2	3	Total
Assigned	8	53	285	346
Unfixed	1	1	8	10
<b>Total</b>	<b>9</b>	<b>54</b>	<b>293</b>	<b>356</b>

- Program Management in action

Status	1	2	3	Total
Postponed	2	49	697	748
Reevaluate		2	16	18
TBA	6	31	314	351
<b>Total</b>	<b>8</b>	<b>82</b>	<b>1027</b>	<b>1117</b>

## Castor Mercury Assessment

Author(s)	Comments	Date
Ana L. & Dan K.	First Draft	10/4/1999
Ana L.	Update	10/8/99, 10/12/99, 10/14/99, 10/19/99, 10/26/99, 10/29/99, 11/2/99, 11/9/99, 11/12/1999, 11/19/99
Ashish	Update	10/20/99, 10/22/99

**Goals for B1** = These goals reflect the quality criteria defined in the MSI Way.

**Required for Phase I Client-Site Testing** = These goals represent risks to the success of client-site testing. They are separate from the B1 goals that need to be met before we can ship Castor with the quality criteria defined in the MSI Way.

	Goals for B1	Current Status	Required for Phase I Client-Site Testing	Comments
<b>Feature Completeness</b> QE signs off on designs, test suites, and feature testing as complete	Server	Y	Y	<b>MD Synch fixed in 600</b> <b>LBD declared Monday 10/11</b>
	Engine	Y	Y	Unit tests and some feature combinations done; further feature combination testing over the next couple of months <b>LBD declared Monday 10/11</b>
	COM	Y	Y	<b>LBD declared Monday 10/11</b>
	Web	Y	Y	<b>LBD declared Monday 10/11</b>
	GUI	Y	Y	Beta Preview scope only for GUI <b>LBD declared by Monday 10/11</b>
<b>Documentation</b> Documentation will be required for Phase II. B1 deliverables include on-line versions and PDF files.	Administrator	Y	N	<b>Files for install available on 10/11. Incorporated in 590. Available through Desktop by 0.0.603 (11/2/99)</b> Item progress: 56%
	Report Designer	Y	N	Item progress: 51%
	Project Designer	Y	N	Item progress: 49%
	Upgrade	Y	N	Item progress: 96%
	Getting Started Guide	Y	N	Item progress: 19%
	Installation and Configuration	Y	N	Item progress: 93%
	Developer (SDK)	Y	N	Item progress: 76%
	Web online help	Y	N	Item progress: 100%, including hooks in interface
	How do I...? online help	Y	N	Item progress: 50%
<b>Stability</b>	Deadlocks = 0	N	N	<b>Known cycles as of 604.2.8 = 3</b> <b>Lock order in 600 (10/25). Due date: 11/20</b> Violations are logged instead of being displayed.
	Endurance Stress Tests	N	N	Automated stress tests should last for 48 hours via the Web with 100 users doing report execution. <b>Best 4-t result: ~ 16 hours.</b> <b>No date estimate for meeting goal. Working on closing gap between automated tests and Web manual tests. New task force in place.</b>
	All-hands stress	N	N	Last one hour with the operations listed in the stress scorecard: 3- and 4-tier. Last test: 141 users, ~800 jobs, 94% CPU usage, 3- and 4-tier <b>On hold until automated stress tests are successful. No date estimate for meeting goal</b>
	Memory leaks and usage = 0	N	N	Leaks reported by the automated tool for 32 scenarios in the scorecard = 100% clean.

	<b>Goals for B1</b>	<b>Current Status</b>	<b>Required for Phase I Client-Site Testing</b>	<b>Comments</b>
				<b>Current values: 100% for leaks, 53% for usage (17 of 32). No date estimate for meeting goal. Entire backend team tackling these tasks.</b>
<b>Performance</b>	Performance Index = 300 rpm	<b>N</b>	N	<b>Current value = 183 rpm, 0.0.604.2.7</b> (309 rpm was achieved w/0.0.600, Date: 10/28/99) Nice to have for site visits
<b>WH Platform certification</b>	Oracle 7.3	Y	Y	
	Oracle 8.0	Y	Y	
	Oracle 8i	Y	Y	
	DB2 UDB 5.2	Y	Y	
	DB/400 V4R3	Y	Y	
	DB2/390	Y	Y	20-30% reports don't run. Engine team investigating – 4 issues, 1 related to timeouts in the DB will remain open. Catalog workaround not implemented, so Server needs to run with a single thread (impacts testing at Marks&Spencer)
	Teradata V2R3	Y	Y	
	Informix OL 7.3	Y	Y	
	SQL Server 7.0	Y	Y	
<b>MD Platform certification</b>	Oracle 7.3	Y	Y	Statistics are not fully implemented.
	Oracle 8.0	Y	Y	
	Oracle 8i	Y	Y	
	DB2 UDB 5.2	Y	Y	
	SQL Server 7.0	Y	Y	
<b>Internationalization</b> Homogeneous environments only GUI will only cover German and Spanish by 10/15 Exploratory testing in Japanese done; no major issues	German	Y	Y	Web fixes confirmed.
	Spanish	Y	Y	All other languages have been tested several times.
	French	Y	N	
	Italian	Y	N	
	Korean	Y	N	Web issue <b>84046 postponed to B2</b>
<b>Installation</b>	Server	Y	Y	IE 5.X details have been fixed. 'Bugfest' happened 10/14 and 10/15 (10/19 for SDK in French, Italian, and Korean)
	Web	Y	Y	
	GUI	Y	Y	
	SDK	Y	Y	
<b>Schema Support</b>	Customer Projects	Y	Y	
	Vmall	Y	Y	
	Internal Beta Sites	Y	Y	
<b>End-to-end story</b>	No S1, S2, high profile S3	Y	Y	Test to be done as part of final regression. Tests on 10/20 revealed no S1 or S2. 11/02 tests produced 1 S2 (resolved), 1 S3
<b>Operating Systems</b>	Windows 95	Y	Y	Exploratory testing done on Win 2000 – Web APS pages couldn't load, under investigation.
	Windows 98	Y	N	
	NT 4.0 SP4	Y	Y	
	Windows 2000	Y	N	
<b>ZDB - TQMS</b>  See notes below	Server	<b>N</b>	N	1 Eng issue open for features. 2 TBC, 17 RTTs
	Engine	N	N	4 open Eng issue (most will be fixed today), 4 TBCs, 5 RTTs
	COM	N	N	ZDB for Eng, 4 TBCs, 13 RTTs
	Web	Y	N	ZDB for Eng and QE
	GUI	<b>N</b>	N	5 Eng in action (unfixed S3s) 10 TBC (S3), 7 RTTs for B1, 46 RTTs overall.

**Beta Programs – Notes from Dan K.**

Based on the lessons of the Alpha program, from talking with Johnetta and given the high expectations our customers have for Castor there are a number of things that must be true before we begin Phase1 Beta testing.

**Required**

- No S1s
- No S2s
- No high profile S3s. This means any S3 that will annoy a customer or give that customer an unduly bad impression of the product. For example, S3s for project creation and upgrade can really hinder testing.
- If there are open S3 issues by the ZDB date, the guidelines for successful site visits, per team, are:
  - Server <= 30
  - COM <= 30
  - Engine <= 30
  - Web <= 30
  - GUI <= 30 (Excluding bugs impacted by feature development)
- A successful regression of the Castor suite. That means we hold at least a two-day regression and then fix all the S1, S2 and high profile S3s that are found.
- The product must be qualitatively stable when running in 3-tier with 1 concurrent user. This is definitely required for internationalized builds of Castor and presently we have not met this standard.

**Highly Recommended**

- Graphs should look good most of the time by default in 3-tier. This is very important for the Europeans who tend to be less tolerant of Beta software.
- Graphs should look the same in 3-tier and 4-tier. (At the very least graphs should look reasonable in 4-tier)
- 2-3 'nice' grid styles for the Web and Agent.
- Agent, although it's only pre-Beta1, should be clean. No obvious re-paint issues, all the appropriate icons and graphics should be in place, etc.

There are 2 key lessons that we should take away from the Alpha program.

1) Better to not go then to go with a bad build. (A bit extreme given that Castor is much more stable than it was a few months ago. However, the expectation for Beta software is also much higher than it was for Alpha software.)

2) Look & feel matter. If there are a few little things that we can do to make the product look more mature it will go a long way towards building momentum and support for the Castor program.

**Castor Beta Programs: Site Overview**

Primary Sponsor Participants	Company	Metadata 6.x	Warehouse	Vertical
Central	Sprint	Oracle 8.0.5.0.0	Oracle 8.0.5.0.0	Telecom
North	Alleghany Ludlum	SQL Server 7.0	Tandem	Manufacturing
North	Estee Lauder	Oracle 7.3	Redbrick 5.1.2	Cnsmr Pkg Goods
North	Warner Lambert	Oracle 8i	Oracle 8i	Cnsmr Pkg Goods
South	Coca Cola	Oracle 8.0.5.0.0	Teradata	
South	First Union Natl. Bank	Oracle 8.0.4.2.1	Oracle 8.0.4.2.1	Banking
South	Glaxo Wellcome	Oracle 7.3	Oracle 7.3	Healthcare/Pharm.
South	Michelin North America	Oracle 7.3	Oracle 7.3	Manufacturing
South	Premier, Inc.	Oracle 7.3	Oracle 7.3	Healthcare/Pharm.
South	USAA	SQL Server 7.0/Oracle	Tandem	Insurance
West (LA)	Earthlink	Oracle 8.0.5.0.0	Oracle 8.0.5.0.0	ISP
West (SF)	Bank Of America	Oracle 7.3	Oracle 7.3	Banking
West (SF)	Visa International	SQL Server 6.5	Informix XPS 8.1	Banking
West (LA)	Western Digital	Oracle	Teradata V2R2	Manufacturing
International				
Int'l (GE)	Bonndata	DB2/UDB	DB2/UDB	Insurance
Int'l (Spain)	La Caixa	Oracle 7.3.3	Oracle 8.1.5	Banking
Int'l (UK)	Marks & Spencer	DB2 MVS	DB2 MVS	Retail
Int'l (GE)	Metro	Oracle	Teradata	Retail
API Beta Program				
North	Estee Lauder	Oracle 7.3	Redbrick 5.1.2	Cnsmr Pkg Goods
South	Glaxo Wellcome	Oracle 7.3	Oracle 7.3	Healthcare/Pharm.
Channels	Lancet			
Channels	Net.Genesis			
Channels	Retek			
Channels	NCR			



## Castor Beta Programs: Site Visit Schedule

Start Date	Region	Company	Beta Consultant
8-Nov	North	Estee Lauder	Sandip Mehta
8-Nov	South	Glaxo Wellcome	John Chon
15-Nov	Central	Sprint	Luis Villafana
15-Nov	West (LA)	Western Digital	Tania Chozet
15-Nov	Int'l (Spain)	La Caixa	Xabier Ormazabal
15-Nov	Int'l (UK)	Marks& Spencer	Arturo Jimenez
6-Dec	Int'l (GE)	Metro	Francesco Biasi
6-Dec	South	First Union Natl. Bank	John Chon
6-Dec	South	Premier, Inc.	Javier Diaz
6-Dec	South	USAA	Luis Villafana
6-Dec	West (SF)	Visa International	Su Yoon
6-Dec	South	Michelin North America	Sandip Mehta
8-Dec	West (LA)	Earthlink	Tania Chozet
8-Dec	Int'l (GE)	Bonndata	Peter Jonnson
Jan-00	North	Alleghany Ludlum	
Jan-00	North	Warner - Lambert	
Jan-00	Int'l (GE)	Filiadata	
Jan-00	West (SF)	Bank Of America	Jeff Rosen
Jan-00	South	Coca Cola	



**Castor Beta Programs: Site Status**

**Late Breaking News:** In some cases the data returned by Castor will have a different level of precision than the data returned by Abell. That often causes the Data Comparison Tool to generate erroneous results. Many of the Beta Consultants have been struggling with this issue and it may be tainting their findings. Further investigation is underway so that we can more accurately represent any issues with data integrity.

**Glaxo Wellcome**

John had a breakthrough of sorts yesterday. After being held up by the engine initialization error for a while, he was able to complete the upgrade by first copying the Abell metadata from Oracle into SQL server and then upgrading from SQL server into Oracle. He was never able to upgrade directly from Abell in Oracle to Castor in Oracle without getting the engine initialization error. He updated Tech Support with this information. The downside is that the "Report Comparison Tool" is saying that about half of the initial 17 reports he tested return different results in Castor than they do in Abell. He is currently investigating to find the cause of the different data on a report by report basis. Technology has been working with John to trouble shoot the problem. A defect has not been logged yet.

**Estee Lauder**

Sandip's P1 (TQMS 86155) forced him to delete a few folders in order to complete the migration. He has now migrated the project, but is finding that many of the reports are returning different data in Castor than they did in Abell. Like John, he is investigating on a report by report basis to see why the data returned is different. Technology is currently working on a fix for this problem.

**Western Digital**

They didn't have the Oracle space ready. It will be ready next week. In the meantime, Tania upgraded the project into SQL server. The reports are running but again we are having data comparison issues between Abell and Castor. 60 of the 92 reports ran returned incorrect data (TS: 77351). Technology has been working with Tania to trouble shoot the problem. A defect has not been logged yet.

**Sprint**

The metadata has been migrated into Castor and reports run. The "Report Comparison Tool" has not yet been run so we don't know if the data comparison problems will be as severe as they have been at other sites.

**La Caixa - Spain**

The initial meetings on Monday were more focused on technical aspects, while the Product Management presentations were scheduled for Friday 19. The customers seem very pleased with the Castor features presented during the demos in both meetings.

The partner company (CP Software) that is in charge of the development and customization through the API also attended the meetings. They are looking forward to meet Xabier again to discuss more technical details.

Status of the upgrade:

Xabier is working on Access database as they haven't assigned yet a database space in Oracle for the upgrade. The DBA is expected to do this next Monday. Object Manager detected some corrupted objects in the MD (reports with missing children) and was not able to fix them. Xabier is working on this.

The upgrade was very fast because the project currently available is very small. Unfortunately about 30% of the objects upgraded reported errors. Xabier is waiting to have the DB space allocated on Oracle to run the upgrade again and investigate more in detail on this.

**Marks and Spencer - UK**

The initial meetings on Monday included the Product Management presentations from Claudio. We received a positive feedback from the customer. They seem pretty excited with the Beta Programs, and they allocated many resources to this.

They have an important project under development and they are willing to roll it out with Castor.  
Status of the upgrade:

Arturo is currently using SQL Server as DB2 MVS is not supported for MD. The upgrade itself went fine but currently there are still problems related to DB2 MVS database space for temp table. Therefore it's not possible to test if the reports are running properly, even if they were upgraded with no errors. The workaround is to allow a single Server thread to connect to the WH.

The problem is that the upgrade was not able to set the DB properties in Castor from the DSS file and due to a bug it's not possible to set these properties manually. tech Support is working on this issue. This sounds like case 77325, which according to the Engine team, can be worked around by doing the following:

When upgrading a project, most settings in the dss file will not be upgraded. If the user wants to add Table Space to the intermediate table names, they should set the Table Space setting in vldbprop.pds. The "Table Space" in vldbprop.pds should be set before you upgrade the project. It will not affect reports in an upgraded project because we do not read from vldbprop.pds when we run reports in an upgraded project. This file is located at Program Files/Common Files/Microstrategy/.

## Castor Resources & Roles

### Engine Resources & Roles

Name	Role	Sub Team/ Responsibility
<b>Management</b>		
Ben Li	CTA	
Jeff Bedell	Program Management	
Ash Jhaveri	Program Management	
Braxton Robbason	Engineering Manager	Query Engine, SQL Engine
Jun Yuan	Engineering Manager	Query Engine, SQL Engine
Xinyi Wang	Engineering Team Lead	Analytical Engine
<b>Analytical Engine</b>		
Yuling Ma	Software Engineer	Analytical Engine
Andrea Torsello	Software Engineer	Analytical Engine
Xiaonan Han	Software Engineer	Analytical Engine
Guanlin Shen	Software Engineer	Analytical Engine
Hani Soewandi	Quality Engineer	Analytical Engine
<b>Query Engine</b>		
Xun Feng	Engineering	Query Engine, Lead
Yi Luo	Software Engineer	Query Engine
Parker Zhang	Software Engineer	Query Engine
Hank Wang	Quality Engineer	Query Engine
Doug Meyer	Advisor/Engineering	Query Engine, Database Classes
Rixin Liao	Engineering	Query Engine
<b>SQL Engine</b>		
Leon Bun	Software Engineer	SQL Engine
Yinong Chen	Software Engineer	SQL Engine, Lead
Sadanand Sahasrabudhe	Engineering Emeritus (Product Management)	SQL Engine
Harinarayan Paramahamsan	Quality Engineer	SQL Engine
<b>Quality</b>		
Lingxiang Chen	Quality Engineer	Lead QE
Jun Shan	Quality Engineer	Customer Projects

**Server Resources & Roles**

Name	Role	Sub Team/ Responsibility	Notes
<b>Engineering</b>			
Wayne Li	Engineering Manager		
<b>Stability and performance team</b>			
Ramprasad Polana	Software Engineering	Technical lead	
Zheng Wang	Software Test Engineering		
Lixin Li	Software Test Engineering		
Abhijit Hayatnagarkar	Software Engineering		In Bootcamp
<b>Execution Flow team</b>			
Ningning Liu	Software Engineering	Technical lead	
Sam Helwig	Software Engineering	Development lead	
Tina Tian	Software Engineering		
Liqun Jin	Software Engineering		Broadcaster integration
<b>XML API team</b>			
Janaki Goteti	Software Engineering	Technical lead	
Yuan Ding	Software Engineering	Development lead	
Ping Xu	Software Engineering		
Yuxiao Xiao	Software Engineering		
Longying Zhao	Software Engineering		
Yi Du	Software Engineering		
<b>Quality Engineering</b>			
Ashish Soni	Quality Engineering	QE lead for all backend teams	
Jianhua Wang	Quality Engineering	QE lead for Kernel team	
Dominique Paschoud	Quality Engineering		XML API
Elsa Polo	Quality Engineering		Execution Flow
Ngone Fall	Quality Engineering		Execution Flow
Sumeet Bhalla	Quality Engineering		Stability and Performance
Hengky Suryadi	Quality Engineering		Acceptance
<b>Documentation</b>			
Randy Hechinger	Tech Writer		
<b>Programs</b>			
Scott Cappiello	Program Manager		
Patrick Vinton	Program Management Engineer	Execution and caching	

**System Component Team**

Name	Role	Sub Team/ Responsibility	Notes
Doug Meyer	Engineering Manager		
Nick Pratt	Software Engineering		
Juan Muraira	Software Engineering		
Andres Murillo	Software Engineering		
Javier Leija	Software Engineering		Testing architecture

**Migration Team**

Name	Role	Sub Team/ Responsibility	Notes
Pat Ori	Programs Engineer	Castor Migration	

***Warehouse Monitor Team***

Name	Role	Sub Team/ Responsibility	Notes
Sascha Naujoks	Warehouse Monitor Engineer	Warehouse Monitor	

***COM Resources & Roles***

Name	Role	Sub Team/ Responsibility
Sean McCafferty	Program Manager	Development team project management.
Will Hurwood	Managing Architect	Overall design and architecture for DSS Objects.
Gary Xue	Development Team Lead	Object Management
Zhiying Chen	Software Engineer	Object Management
Cezary Rascko	Software Engineer	Object Management
Jing Li	Development Team Lead	Object definitions schema and application.
Dan Preotescu	Software Engineer	Object definitions and parser development
Ian Falicov	Software Engineer	Object definitions.
Fabian Camargo	Development Team Lead	Element Browsing, Prompting, Report Resolution
Ozgur Huseyinoglu	Software Engineer	
Glenn Boysko	Manager	SDK program management and test engineering
Yansong Wang	Quality Engineer	Object Management, Prompting, Element Browsing
Peter Hefner	Documentation	Developer Guide and API Specification
Jitendra Shirolkar	Software Test Engineer	DSS Web 5.x API Customer Migration, Web API Testing
Lawrence Lun	Software Test Engineer	Drilling, SDK Test Framework/Infrastructure
Lixin Shou	Software Test Engineer	XML Validation (all forms)
Chen Qian	Software Engineer	Application Engineering--Sample Applications
Fernando Gonzalez	Quality Engineer	TQMS Management, Regression Tests, Acceptance Tests

***Desktop Resources & Roles***

Name	Role	Sub Team/ Responsibility
Fabrice C. Martin	Program Manager	Castor GUI program management
Eduardo Carranza	Engineering Manager	Overall engineering management
Arturo Gay	Engineering Manager	Administration GUI management & engineering
Erika Kuswa	QE Manager	Castor GUI Quality Engineering management
Javier Aldrete	Engineer	Castor Architect Editors design and engineering
Sudhakar Nelamangala	Engineer	Filter Editor & Castor GUI Engineering and design
Jing Ning	Engineer	Administration tools and dialogs design and engineering
Andres Paz	Engineer	Metric Editor & Castor GUI Engineering and design
Sergio Trejo	Engineer	Object Browser and Castor GUI design and engineering
Adel Elcheik	Quality Engineer	Metric and Filter functionality quality engineering
Olivia Moncayo	Quality Engineer	Castor Architect quality engineering
Roya Khaizard	Quality Engineer	Desktop viewers quality engineering
Chaitan Kansal	Software Test Engineer	Castor GUI quality engineering
Victor Peña	Engineer	Desktop Viewers
Jorge Garcia	Engineer	Schema Printing Component
Mayra Madrigal	Quality Engineer	Application level editors quality engineering
Hector Aguilera	Quality Engineer	Castor Architect quality engineering
Carlos Madrid	Quality Engineer	Object Browser quality engineering
Quyen Diep	Quality Engineer	Quality Engineering
Ji Jin	Engineer	Viewers design and engineering
Iracly Kakushadze	Engineer	Viewers design and engineering
Brian Shanahan	Quality Engineer	Automated testing design & development

***Web Resources & Roles***

Name	Role
Doug Everhart	Program Manager
Gunther Brenes	Software Architect, GUI Design
Arturo Oliver	Web GUI Design & implementation, Engineering Manager
Jiefeng Li	Web GUI Design & implementation (XSLs)
Jupiter Munoz	Web GUI Design & implementation (asp/XSL)
Victor Arjona	Software Engineer
Andrew Smith	QE Lead
Alda Cheng	Quality Engineer
Jonathan Jiang	QE

***Extended QE Resources & Roles******QE Integration Team***

Name	Role
Olivier Marchal	QE Lead
Mala Viswanath	Installation, configuration wizard, and diagnostics (for Beta 1)
José Rosas	End-to-end story and installation

***QE Cross Team***

Name	Role
Ana López	QE Release Manager
Dan Kerzner	Alpha and Beta Programs Lead
Cuong Bui	VMALL Engineer
Shandee Chernow	Internal Beta Sites Engineer
Srinivas Rayarao	Acceptance QE

***QE Systems Analysis team***

Name	Role
AnneMarie Ferraro	QE Manager
Mario Guagnelli	QE Lead, Enterprise
Yi Liu	Server, EA
Jorge López	Server, EA
Pankaj Bengani	Web, EA
Claudia Rodríguez	GUI, EA
Sheila Somani	Usability
Jeanette Chian	Usability
Florence Lu	Usability
Nat Venkataraman	CAG
Plinio de los Santos	CAG, Global
Benny Sukamto	CAG



## Castor Program Status – 12/23/99

Status.....	2
Overall Summary .....	2
Team highlights.....	3
Project Progress.....	5
High-level Schedule .....	5
Cross-Team Release Criteria.....	5
QE Assessment .....	6
QE Metrics .....	6
Risk Assessment.....	6
Beta 1 Sign Off Issues .....	6
QE Team Highlights.....	8
Beta Program Summary.....	10
Resources & Roles .....	12
Engine Resources & Roles .....	12
Server Resources & Roles.....	12
COM Resources & Roles .....	13
Interface Resources & Roles .....	14
Web Resources & Roles .....	14
SDK Team .....	15
Cross-Product QE Teams .....	15

## Status

### *Overall Summary*

The happiest news of the past month was the Castor Team's achievement of the Beta 1 milestone. On Tuesday, December 13, we announced this significant achievement, which was culminated by formal signoff of the quality of the product. In order to achieve the Beta 1 milestone, the product had to:

- Pass a 48 hour automated regression test
- Survive multiple all-hands stress tests
- Achieve a score of 250 reports per minute on the I-Benchmark
- Resolve all known open Severity 1, Severity 2, or Severity 3 defects, subject to the formal sign-off from a broad set of technology managers who will see that any unresolved issues are addressed prior to General Availability.

The final barrier that prevented the team from reaching the Beta 1 milestone earlier was a set of stability and performance objectives defined for the product suite as a whole. In the weeks leading up to the Beta 1 release, the backend teams focused on these stability objectives while the frontend teams were able to make progress on Beta 2 plans.

After going through the Beta 1 signoff process, we recognized the need for more frequent milestones as we progress towards GA. The signoff documents included an alarming number of known issues, many of which are pending additional development work. We need to make a serious effort to manage the risk associated with these issues staying open in the Beta 1 release.

As a result, we have reorganized our development path to GA to consist of more frequent milestones. The individual teams are arranging their plans so that we can issue a series of monthly builds that are each stable and of good quality. The team is currently driving towards our January milestone, which will essentially bring the frontends and backend back together and also include issue fixes.

The team is also driving towards a set of defined release criteria. This set of objectives includes targets for stability, scalability, performance, and other cross-team desiderata. This model worked for the drive towards the Beta 1 objective, so we are extending it for the GA release. This method provides a useful set of management controls; as we release each monthly build, we'll know how close we are to GA.

In the meantime, the Beta Program continues to exercise the product at customer sites. We are currently working with 16 partners and customers as part of the first phase of the Beta program. A group of Beta consultants are devoted full-time to working on these sites. Our strategy is to ensure referenceable success with this targeted set of customers. We will also open up the Beta program to a broader audience in the second phase of the Beta program. We expect this to coincide with the March build milestone and it will include 60-70 new participants.

We have also made progress on our so-called "internal Beta" projects. The internal beta team has published to Technology and Product Management the URL to a Castor-powered version of the Corporate Development site. The team has also developed the Product Education Portal (PEP) which showcases the full functionality of the Castor release. The PEP project is the centerpiece of this month's demonstration and we plan to enhance it and unveil it for the rest of the company on the cruise.

Finally, a large part of the Castor Team participated in Technology's University week. Another substantial portion of the team will participate in week 2 of University Week after the cruise.

As we go out to customer sites with the product, we are seeing more and more demand for services associated with the release, such as training and project guidance through the migration process. We look forward to working with other functions within the company as everyone begins to prepare for and rally around the Castor release.



***Team highlights*****Web**

- Working with Olympus Group to redesign the Web GUI to be more consistent with Strategy.com.
- Continuing to work on Beta 2 changes.
- With the recent changes to the Beta release schedule, the Web team will focus from now until the Cruise on stabilizing the Beta 2 branch to ensure that it meets or exceeds the quality criteria for a Beta release.

**Desktop**

- Worked for the most part on Bug Fixing & Stabilization tasks
- Completed Impact Analysis functionality for all Schema Objects
- Completed designs and preliminary implementation for Report Export functionality
- Completed designs and implementation for Custom Subtotaling and Report Sorting

**Server****COM**

- Bug fixing
- Beta 2 Planning – Consists mostly of bug fixing and performance enhancements. A few small feature improvements remain:
  - Filter Details – Need to cleanup representation and verify content
  - Number Formatting
    - Consolidate 3T and 4T usage.
    - Add addition % recognition through parsing
    - Add fraction support
    - Evaluate 3<sup>rd</sup> party formatting libraries
  - New XML DOM – eliminate overhead and performance constraints of Microsoft's DOM. XML generation is currently a major performance bottleneck.
  - Prompt on TemplateCustomGroup (NetGenesis request)
  - Linked Properties cache – This should improve the performance and scalability of our linked properties which allows for user personalization. The linking of users to specific DSS Objects.
- Documentation of final interfaces

**Kernel**

- Focused our efforts on achieving Beta 1 performance and stability goals:
  - Eliminating potential deadlocks from the system.
  - Reduce memory leaks and memory usage for key operations.
  - Stress the server with automated tools in a 48-hour endurance test.
  - Stress the server with manual usage in all-hands tests.
  - Achieve a score of 250 rpm on the I-Benchmark.
- Refined release criteria for second phase of Beta and GA.
- Developed initial plans in pursuit of those criteria.
- Participated in University week.

**SDK**

- Enhanced the diagnostic Web site to add new features such as, drilling, searching, and element browsing.
- Initiated work on the Migration tool kit, Extended Properties Editor, User Manager for Web, and Function Server Plug-In.
- Participated in the IBM mentored workshop.

- IBM WebSphere Phase II investigation progress includes, upgrading to Web Sphere 3.02, and migrating key pieces of DSS Web GUI from ASP to JSP. The key pieces involve login, object browsing, and report execution.
- Visited Estee Lauder and Lancet. Provided them with the Web API training course and discussed functionality migration issues.
- Met with ProVantage and net.Genesis to discuss Castor COM and Castor Web API.
- Provided Web API training during University week.

## Project Progress

### High-level Schedule

Milestone	Description	Duration	Start	Finish
Beta 2	Merge frontend feature changes with backend bug fixes	6 weeks	Mon 12/20/99	Fri 1/28/00
Beta 3	Remaining feature development, bug fixing, stabilization.	4 weeks	Mon 1/31/00	Fri 2/25/00
Beta 4	Expected build to use for Phase II of the Beta Program	5 weeks	Mon 2/28/00	Fri 3/31/00
Beta 5	Bug fixing and stabilization.	4 weeks	Mon 4/03/00	Fri 4/28/00
RC	Release candidate	2 weeks	Mon 5/01/00	Mon 5/15/00
GA				Tue 5/16/00

### Cross-Team Release Criteria

Area	Objective	Goal
Features	Test suites	Complete testing and QE sign-off for scoped features
TQMS	ZDB	Meet MSI-Way ZDB criteria for a Beta/GA release
Cross-product	Documentation	Complete hard-copy documentation
	WH Platforms	Certify all scoped RDBMS's
	MD platforms	Certify all scoped RDBMS's
	OS platforms	Certify all scoped operating systems
	International environments	Certify all international requirements
	Diagnostics	Complete testing and certify compliance with Diagnostics spec
	End-to-end story	Complete and pass end-to-end, cross-product feature test
	Installation	Certify installation routines
Stability	Stress acceptance test	Pass daily acceptance test
	Endurance test on customer site	Pass overnight stress test on site
	Endurance test in-house	Pass 20000 users, 5% concurrency, 2 million jobs test for 48 hours on clustered configuration
	4-tier all-hands stress	Pass stated test, including performance expectations on clustered configuration
	Boundary cases	Finish stated list of tests
Scalability	Mem usage acceptance test	Pass daily acceptance test
	Footprint analysis	Finish stated list of tests, including desktop and backend analysis
Performance	Performance acceptance test	Pass daily acceptance test
	3-tier single-user response time	Surpass response time index for 3-tier operations with single user
	4-tier multi-user response time	For 4-tier operations, ensure 8 s response time for typical operations and 3 s response time for key operations, with 300 users load on server
Successful Applications	15 external beta sites	15 external sites of which 3 are new projects must be referenceable
	5 internal beta sites	5 internal sites must be production ready-- production readiness to be determined beforehand by executives
Engine	SQL Execution Time - Internal	For customer projects, complete performance comparison and ensure that new Castor reports are faster than Abell
	SQL Execution Time - External	Complete the performance comparison using new Castor reports on 15 beta sites
	Data Accuracy	For 15 Beta sites, 99% of Castor reports will match Abell results within acceptable tolerance for precision
	In-house customer projects	Finish stated list of tests, e.g. upgrade and run all reports for 10 metadatas in house
	SQL Execution	99% of reports from 15 Beta sites must execute (i.e. SQL Can be generated)
Strategy.com	Analytic sophistication	Ensure that the Castor Engine supports the analytical requirements of Strategy.com's Investment Channel

## QE Assessment

### QE Metrics

- Number of issues open for Mercury GA: We have **1036** issues open for engineering, **172** issues open for Program Management, and **315** issues open for QE.

QE in action

Status	1	2	3	Total
Ready to Test	1	20	219	240
TBC by QE	0	5	70	75
<b>Total</b>	<b>1</b>	<b>25</b>	<b>289</b>	<b>315</b>

Engineering in action:

Status	1	2	3	Total
Assigned	5	82	945	1032
Unfixed	0	0	4	4
<b>Total</b>	<b>5</b>	<b>82</b>	<b>949</b>	<b>1036</b>

Program Management in action:

Status	1	2	3	Total
Postponed	0	0	19	19
Reevaluate	0	0	6	6
TBA	1	19	127	147
<b>Total</b>	<b>1</b>	<b>19</b>	<b>152</b>	<b>172</b>

### Risk Assessment

#### Major risks going forward

- Database Classes will not be in Beta 2. This is a major change and delaying this into the main build may lead to high severity issues being uncovered at a late stage.
- Need to expand test coverage. As an example :- our Beta 1 memory leak/usage scenarios showed 0 memory leaks. However in an all hands stress environment large memory usage is observed.
- Supporting Beta sites adequately. We've defined procedures and have been working with Tech Services to have as many of the client projects in-house as possible, to alleviate this risk. However, when we get to Beta 2, it will become an open enrollment program, with up to 70 clients signed on.
- Number of issues open for Mercury GA: We already have **1036** issues open for engineering, **172** issues open for Program Management, and **315** issues open for QE.

### Beta 1 Sign Off Issues

#### Web

Issue Type	1	2	Total
Defects	0	3	3
Feature Development	0	0	0
Enhancement	0	3	3
<b>Total</b>	<b>0</b>	<b>6</b>	<b>6</b>

***Cross Team***

<b><i>Issue Type</i></b>	<b><i>1</i></b>	<b><i>2</i></b>	<b><i>Total</i></b>
Defects	0	3	3
Feature Development	0	0	0
Enhancement	0	0	0
<b><i>Total</i></b>	<b><i>0</i></b>	<b><i>3</i></b>	<b><i>3</i></b>

***Engine***

<b><i>Issue Type</i></b>	<b><i>1</i></b>	<b><i>2</i></b>	<b><i>Total</i></b>
Defects	1	37	38
Feature Development	0	6	6
Enhancement	1	13	14
<b><i>Total</i></b>	<b><i>2</i></b>	<b><i>56</i></b>	<b><i>58</i></b>

***SDK***

<b><i>Issue Type</i></b>	<b><i>1</i></b>	<b><i>2</i></b>	<b><i>Total</i></b>
Defects	0	0	0
Feature Development	0	0	0
Enhancement	0	0	0
<b><i>Total</i></b>	<b><i>0</i></b>	<b><i>0</i></b>	<b><i>0</i></b>

***GUI***

<b><i>Issue Type</i></b>	<b><i>1</i></b>	<b><i>2</i></b>	<b><i>Total</i></b>
Defects	0	7	7
Feature Development	0	0	0
Enhancement	0	0	0
<b><i>Total</i></b>	<b><i>0</i></b>	<b><i>7</i></b>	<b><i>7</i></b>

***Server***

<b><i>Issue Type</i></b>	<b><i>1</i></b>	<b><i>2</i></b>	<b><i>Total</i></b>
Defects	0	3	3
Feature Development	0	29	29
Enhancement	0	21	21
<b><i>Total</i></b>	<b><i>0</i></b>	<b><i>53</i></b>	<b><i>53</i></b>

***COM***

<b><i>Issue Type</i></b>	<b><i>1</i></b>	<b><i>2</i></b>	<b><i>Total</i></b>
Defects	1	11	12
Feature Development	0	0	0
Enhancement	0	4	4
<b><i>Total</i></b>	<b><i>1</i></b>	<b><i>15</i></b>	<b><i>16</i></b>

## ***QE Team Highlights***

### ***CrossTeam***

#### **Highlights:**

- We are Beta1! :) Let's enjoy the moment even though the finish line is still far away...
- We have had only one case logged about the Castor install during the Beta Programs so far (which turned out to be an issue from PC Anywhere)
- We are currently setting the roadmap structure in terms of diagnostics implementation for the Web, Server and GUI teams
- We are preparing an SQA class for miniversity to train our testers in front-end automation
- We have been reorganizing the build room to gather all the team build machines and acceptance machines. This will allow us to better overview our processes and work as a team

#### **Lowlights:**

- We need to increase the coverage of the acceptance, and we are not fast enough in doing this.

### ***Web***

#### **Highlights:**

- getting a good start with B2. (are we allowed to say that?? )B2 pages
- are approaching stability.
- we now have a 6-ppl castor web qe team, new people are getting up to speed.
- chongyan gave birth to a boy.
- from visa: "The customer loves HTML web. They have been waiting for Castor Web for a web rollout. In the future they expect 90% of their users to be through the Web."

#### **Lowlights:**

- code-changes to improve performance still needs to be tested.
- we do not have much automation in testing, but as Arvind and chongyan return, we will expect to see much of that in place in Jan and Feb.

### ***GUI***

#### **Highlights:**

- Our University week will focus on automation. We expect this to improve our test coverage, and possibly the time it takes to run regression.

#### **Lowlights:**

- We have a very high number of severity 3 defects to clean out before GA. QE will have to help prioritize these in order to get the worst ones fixed first.

### ***Server***

#### **Highlights:**

- We are Beta.
- Finished Automated 50% of the 3T testing and 95% of the 4T API testing.
- Meet with APS and Beta consultant on the weekly basis, greatly improved the communication.
- Diagnostics turned out to be THE most useful tool to trouble shoot issues found in the client site.
- Half of the team participated technology University week, gained a lot of new knowledge about the product.

- Team building: We have built a very knowledgeable Kernel QE team with 3 people specialized in stress testing, 3 people specialized in castor web API, everyone can write VB testing program, and for each server feature there are at least 2-3 people have expertise.

**Lowlights:**

- A lot of issues found in the client side doesn't need to reach technology if we disseminate our knowledge to APS! Communication between TS and our need to be improved even further.
- We haven't done a lot of stress test in cluster environment.

***Engine*****Highlights:**

- Engine team passed Beta 1 sign off;
- Engine QE has adjusted customer projects testing strategy to make the test much more effective;
- Engine QE has finished the test case review of all major Engine features;

**Lowlights:**

- We realized that we do not have a good way to measure the Engine quality and present it to the outside world;
- Engine features are getting out of control, i.e., we allow people to build many kinds of reports that either break or return ridiculous results. We are addressing this issue;

***COM*****Highlights:**

- Got the Beta 1 document signed off!
- Test Framework in place, and COM Developers help to write testing code.
- TQMS issues under control.
- Start work with Engine team on Engine Random Report test.

**Lowlights:**

- Many issues come from field test

**Beta Program Summary**

<b>Table of Site Progress</b>				
<b>Client Site</b>	<b>Beta Consultant</b>	<b>Date Started</b>	<b>Progress Level</b>	<b>Is the Consultant Currently On this Site?</b>
Estee Lauder	Sandip Mehta	11/08/99	5	No (at Michelin)
Glaxo Wellcome	John Chon	11/08/99	6	No (at First Union)
La Caixa	Xabier Ormazabal	11/15/99	5	Yes
Marks and Spencer	Arturo Jimenez	11/15/99	6	Yes
Sprint	Luis Villafana	11/15/99	5	No (at USAA)
Western Digital	Tania Chozet	11/15/99	6	Yes
Bonndata	Peter Jonsson	12/06/99	5	Yes
First Union	John Chon	12/06/99	6	Yes
Metro	Francesco Biasi	12/06/99	5	Yes
Michelin	Sandip Mehta	12/06/99	5	Yes
Premier	Javier Diaz	12/06/99	5	No (on Vacation)
USAA	Luis Villafana	12/06/99	6	Yes
Visa	Jeff Rosen	12/07/99	6	Yes
net.Genesis	Su Yoon	12/13/99	3	Yes

<b>Scale of Site Progress</b>		
<b>Level</b>	<b>Name</b>	<b>Description</b>
1	Presentation and Demonstration	The client has attended an on-site MicroStrategy 7 presentation and demonstration that was conducted by the Beta Consultant and Product Management.
2	Project Plan Creation	The Beta Consultant and the client have discussed the MicroStrategy 7 project plan that the Beta Consultant intends to implement before the end of Beta One.
3	Product Installation	The Beta Consultant has successfully installed the MicroStrategy 7 software in the client's test environment and has secured the necessary database space and permissions for testing.
4	Metadata Migration (Upgrade)	The Beta Consultant is in the process of migrating the client's project from the old MicroStrategy 6 metadata format to the new MicroStrategy 7 metadata format.
5	Migration Acceptance	The Beta Consultant is ensuring that the metadata migration was successful. A representative sample of reports must be run from the project in both the original MicroStrategy 6 architecture and the new MicroStrategy 7 architecture. Any discrepancies in the results must be examined, logged, and troubleshooted.
6	Test Suite Execution (Front End)	The Beta Consultant has a stable MicroStrategy 7 project on the client site and is now performing front end related test suites. Among these test suites are the GUI test suites and the Web interface test suites.
7	Test Suite Execution (Back End)	The Beta Consultant has a stable MicroStrategy 7 project on the client site and is now performing back end related test suites. Among these test suites are the Server test suites including stress testing on selected sites.



8	Project Plan Implementation (Phase 1)	The Beta Consultant is implementing the first major feature in the MicroStrategy 7 project plan for the client. The actual feature being implemented varies from site to site. See a site's full weekly update document for more information on its site specific project plan.
9	Project Plan Implementation (Phase 2)	The Beta Consultant is implementing the second major feature in the MicroStrategy 7 project plan for the client. The actual feature being implemented varies from site to site. See a site's full weekly update document for more information on its site specific project plan.
10	Visit Completion	The client has a fully functional MicroStrategy 7 project in the development environment which includes advanced MicroStrategy 7 functionality that has been implemented by the Beta Consultant.

## Resources & Roles

### Engine Resources & Roles

Name	Role	Sub Team/ Responsibility
<b>Management</b>		
Ben Li	CTA	
Jeff Bedell	Program Management	
Ash Jhaveri	Program Management	
Jun Yuan	Engineering Manager	Query Engine, SQL Engine
Xinyi Wang	Engineering Manager	Analytical Engine
<b>Analytical Engine</b>		
Yuling Ma	Software Engineer	Analytical Engine
Xiaonan Han	Software Engineer	Analytical Engine
Guanlin Shen	Software Engineer	Analytical Engine
Hani Soewandi	Quality Engineer	Analytical Engine
Rui Gong	Quality Engineer	Analytical Engine
<b>Query Engine</b>		
Xun Feng	Engineering	Query Engine, Lead
Yi Luo	Software Engineer	Query Engine
Parker Zhang	Software Engineer	Query Engine
Hank Wang	Quality Engineer	Query Engine
Rixin Liao	Engineering	Query Engine
Hai Sho	Software Engineer	Query Engine
<b>SQL Engine</b>		
Leon Bun	Software Engineer	SQL Engine
Yinong Chen	Software Engineer	SQL Engine, Lead
Harinarayan Paramahamsan	Quality Engineer	SQL Engine
<b>Quality</b>		
Lingxiang Chen	Quality Engineer	Lead QE
Jun Shan	Quality Engineer	Customer Projects
Piyali Dey	Quality Engineer	

### Server Resources & Roles

#### Kernel Team

Name	Role	Sub Team/ Responsibility
<b>Engineering</b>		
Wayne Li	Engineering Manager	
<b>Stability and performance team</b>		
Ramprasad Polana	Software Engineering	Technical lead
Zheng Wang	Software Test Engineering	
Lixin Li	Software Test Engineering	
Abhijit Hayatnagarkar	Software Engineering	
<b>Execution Flow team</b>		
Ningning Liu	Software Engineering	Technical lead
Sam Helwig	Software Engineering	Development lead
Tina Tian	Software Engineering	
Liquan Jin	Software Engineering	Broadcaster integration

<b><i>XML API team</i></b>		
Janaki Goteti	Software Engineering	Technical lead
Yuan Ding	Software Engineering	Development lead
Ping Xu	Software Engineering	
Yuxiao Xiao	Software Engineering	
Yi Du	Software Engineering	
<b><i>Quality Engineering</i></b>		
Ashish Soni	Quality Engineering Manager	QE manager for all backend teams
Jianhua Wang	Quality Engineering	QE lead for Kernel team
Dominique Paschoud	Quality Engineering	XML API
Elsa Polo	Quality Engineering	Execution Flow
Ngone Fall	Quality Engineering	Execution Flow
Sumeet Bhalla	Quality Engineering	Stability and Performance
Hengky Suryadi	Quality Engineering	Acceptance
<b><i>Documentation</i></b>		
Randy Hechinger	Tech Writer	

***System Component Team***

Name	Role	Sub Team/ Responsibility
Doug Meyer	Engineering Manager	
Nick Pratt	Software Engineering	
Juan Muraira	Software Engineering	
Andres Murillo	Software Engineering	
Javier Leija	Software Engineering	Testing architecture

***Programs***

Name	Role	Sub Team/ Responsibility
Scott Cappiello	Program Manager	
Patrick Vinton	Program Management Engineer	Execution and caching features
Pat Orie	Program Management Engineer	Castor Migration
Sascha Naujoks	Programs Associate	PEP project development, Warehouse Monitor research

***COM Resources & Roles***

Name	Role	Sub Team/ Responsibility
<b><i>Engineering</i></b>		
Will Hurwood	Managing Architect	Overall design and architecture for DSS Objects.
<b><i>Object Management Team</i></b>		
Gary Xue	Development Team Lead	Object Management
Zhiying Chen	Software Engineer	Object Management
Cezary Razcko	Software Engineer	Object Management
<b><i>Parser Team</i></b>		
Dan Preotescu	Development Team Lead	Object definitions and parser development.
<b><i>DSS Objects Definitions</i></b>		
Jing Li	Development Team Lead	Object definitions schema and application
Ian Falicov	Software Engineer	Diagnostics, Object definitions.
Yasser Mufti	Software Engineer	Log Viewer

<b>Prompt Resolution and Element Browsing</b>		
Fabian Camargo	Development Team Lead	Element Browsing, Prompting, Report Resolution
Ozgur Huseyinoglu	Software Engineer	XML generation for element browsing, prompting, searching.
Harpreet Duggal	Software Engineer	Save As functionality
<b>Other</b>		
Gary Anderson	Software Engineer	New.
Yuesong Wang	Test Engineer	COM unit tests and drivers.
<b>Quality Engineering</b>		
Yansong Wang	Lead COM QE	
Fernando Gonzalez	QE	
Nilesh Gandhi	QE	
<b>Programs and Project Management</b>		
Sean McCafferty	Program Manager	

### Interface Resources & Roles

Name	Role	Sub Team/ Responsibility
Fabrice C. Martin	Program Manager	Castor GUI program management
Eduardo Carranza	Engineering Manager	Overall engineering management
Arturo Gay	Engineering Manager	Administration GUI management & engineering
Erika Kuswa	QE Manager	Castor GUI Quality Engineering management
Javier Aldrete	Engineer	Castor Architect Editors design and engineering
Sudhakar Nelamangala	Engineer	Filter Editor & Castor GUI Engineering and design
Jing Ning	Engineer	Administration tools and dialogs design and engineering
Andres Paz	Engineer	Metric Editor & Castor GUI Engineering and design
Sergio Trejo	Engineer	Object Browser and Castor GUI design and engineering
Adel Elcheikh	Quality Engineer	Metric and Filter functionality quality engineering
Olivia Moncayo	Quality Engineer	Castor Architect quality engineering
Chaitan Kansal	Software Test Engineer	Castor GUI quality engineering
Victor Peña	Engineer	Desktop Viewers
Jorge Garcia	Engineer	Project Upgrade & Duplication
Mayra Madrigal	Quality Engineer	Application level editors quality engineering
Hector Aguilera	Quality Engineer	Castor Architect quality engineering
Carlos Madrid	Quality Engineer	Object Browser quality engineering
Quyen Diep	Quality Engineer	Quality Engineering
Ji Jin	Engineer	Viewers design and engineering
Iracly Kakushadze	Engineer	Viewers design and engineering
Juan Ontiveros	Engineer	Castor Object Manager

### Web Resources & Roles

Name	Role	Sub Team/ Responsibility
Arturo Oliver	Engineering Manager	
Gunther Brenes	Software Architect, GUI Design	
Luis Dector	Lead Software Engineer	
Jiefeng Li	Software Engineer	
Jupiter Munoz	Software Engineer	
Victor Arjona	Software Engineer	
Wenging Deng	Software Engineer	
Nader Akhnoukh	Software Engineer	

Andrew Smith	QE Lead	
Alda Cheng	Quality Engineer	
Daniel Esparza	Quality Engineer	
Chongyan Huo	Quality Engineer	
Rachel Kern	Quality Engineer	
Arvind Narayanaswamy	Quality Engineer	
Doug Everhart	Program Manager	
Kate Hersey	Program Manager Associate	

***SDK Team***

Name	Role	Sub Team/ Responsibility
Glenn Boysko	Development Manager	
Jitendra Shirolkar	Senior Software Engineer	Web migration toolkit/Sample application development/Customer training and migration support.
Lixin Shou	Software Design Engineer	Web and COM API based web samples and test applications.
Qian Chen	Software Design Engineer	Web API based web samples/COM API based samples.
Lawrence Lun	Software Design Engineer	Test applications and Web API based samples.
Kevin Maurer	Software Engineer	IBM Websphere investigation.
Craig Silverstein	Principal Quality Engineer	Web migration toolkit. On temporary rotation from QE.
Cupid Chan	Software Quality Engineer	Testing sample applications. Under training.
Peter Hefner	Senior Technical Writer	Documentation.

***Cross-Product QE Teams******QE Release Management Team***

Name	Role
Ana López	QE Release Manager
Dan Kerzner	Alpha and Beta Programs Lead
Cuong Bui	VMALL Engineer
Shandee Chernow	Internal Beta Sites Engineer
Srinivas Rayarao	Acceptance QE

***QE Integration Team***

Name	Role
Olivier Marchal	QE Lead
Mala Viswanath	Installation, configuration wizard, and diagnostics (for Beta 1)
José Rosas	End-to-end story and installation

***QE Systems Analysis team***

Name	Role
Mario Guagnelli	QE Lead, Enterprise
Yi Liu	Server, EA
Jorge López	Server, EA
Pankaj Bengani	Web, EA
Claudia Rodríguez	GUI, EA
Sheila Somani	Usability
Jeanette Chian	Usability
Florence Lu	Usability
Nat Venkataraman	CAG
Plinio de los Santos	CAG, Global
Benny Sukanto	CAG

# Deliverables by Week

2/12

*SM - Project Upgrade*

2/19

No deliverables

2/26

*SM – Embedded Objects*

*WL - Report Subsetting: Filter Subsetting*

3/5

*XW – Nested Aggregation – Resolution of Dimty*

*XW – Nested Aggregation – Expression Evaluation*

*XW – Olap functions – Applying sort*

*XW – Olap functions – converting sort*

*WL - Access control, privilege checking on server operations*

*EC – Custom Groups – SQL Engine – Conversion BE*

3/12

*XW – Nested Aggregation – Resolution of rcp*

*SC - Job Prioritization*

*WL - Enhancement in report execution cycle*

*RR – Login/Authentication*

*RR – Object Browsing*

3/19

*SC - Schedule Definition*

*EC - Object Browser Improvements*

*EC - Populating Custom Group BE*

*EC – Relationship Filter BE*

*EC - Toggle between Grid and Graph - Abell  
Functionality*

*RR – Report Execution*

3/26

*SS - Partitioning – Heterogeneous/Incongruous*

*JY – Null/Zero Handling*

*SC - SERVER ADMIN - Security: Application Access*

*SC - SERVER ADMIN - VLDB Properties*

*WL – XML API - History-list, Inbox*

*EC – Consolidations - Backend Integration - SQL  
and Analytical Engine*

*EC – Custom Groups - Custom Group Population  
Instruction BE*

4/2

*SC - SERVER ADMIN - Cluster Admin*

*WL - Caching Enhancements*

*EC - Attribute Editor*

*EC – Custom Groups - Banding Operator BE*

*EC – Custom Groups - SQL Engine*

4/9

MW – Element Browsing - Web Support  
SC – SERVER ADMIN - Scheduling Administration  
SC - SERVER ADMIN - Database Objects  
SC – Config Wizard - Sample DB Creation  
WL - Backup/restore inbox messages  
WL – XML API – Element Browsing  
EC - Hierarchy Editor  
EC – Custom Groups COM  
RR – Element Browsing (w/ incremental fetch)  
RR - Inbox  
RR – export to excel

4/16

SS - Distributed Databases – Abell Functionality  
SS - VLDB - Syntax Pattern Abstraction  
JY - SQL Cancel  
JY – Governing  
JY - Nested Agregation (SQL engine only)  
XW - Compound Metrics - Integration with consolidation  
XW – Olap functions – OLAP function in AE  
SC - SERVER ADMIN - Caching: Admin and Monitoring  
SC - SERVER ADMIN - Helper applet  
WL – Load Balancing and Fail Over  
WL - Web Server API support  
EC - Metric Expression Qualification BE



EC – Filter SQL Engine  
EC – Metrics – Usability improvements  
EC – Report Editor – Usability improvements  
RR – Graphs  
RR – sorting  
RR – Pivoting  
RR – Page by

4/23

SS - VLDB – RDBMS First Class Object  
JY - Count/Rank Consider NULL  
JY - Analytical Function on Fact  
JY - SQL Function Type  
MW - Web Prompt Support  
MW – Authentication  
MW - User Group Management  
WL - Incremental Fetch  
WL - Session Manager  
WL - Session Manager to support Incremental  
fetching  
WL - Administration Monitoring and Statistics -  
Cache Admin and Monitoring  
EC – Folder Reorganization  
EC – Project Manager - Project Upgrade and  
Duplicate  
RR - Prompting

4/30

MW - MD Security  
MW - Access Control  
SM - Linked Properties  
XW - OLAP Function Support in SQL Engine  
XW - Compound Metrics - Smart Totalling  
MW – Drilling BE  
WL - Session based history list/inbox  
WL – XML API – Prompts  
WL – Web Server - Modify all output messages into XML  
WL - Gif file generation for Report  
RR - Document

5/7

JY - OLAP Function of RDBMS  
JY - Catalog Lock Workarounds  
JY - Total Dimension VA  
WL - Document Processing  
EC – Logon Improvements  
EC – Metric Editor  
EC – Grid  
RR - Drilling

5/14

MW – Support for Date time prompts  
WL – XML API - Grid transformation: Pivot, Outline, Sorting  
EC – Custom Groups End to End

EC – Project Manager - Project Creation and Update  
EC - Integration of ThreedGraphics and MSI  
RR – default desktop  
RR – Printing